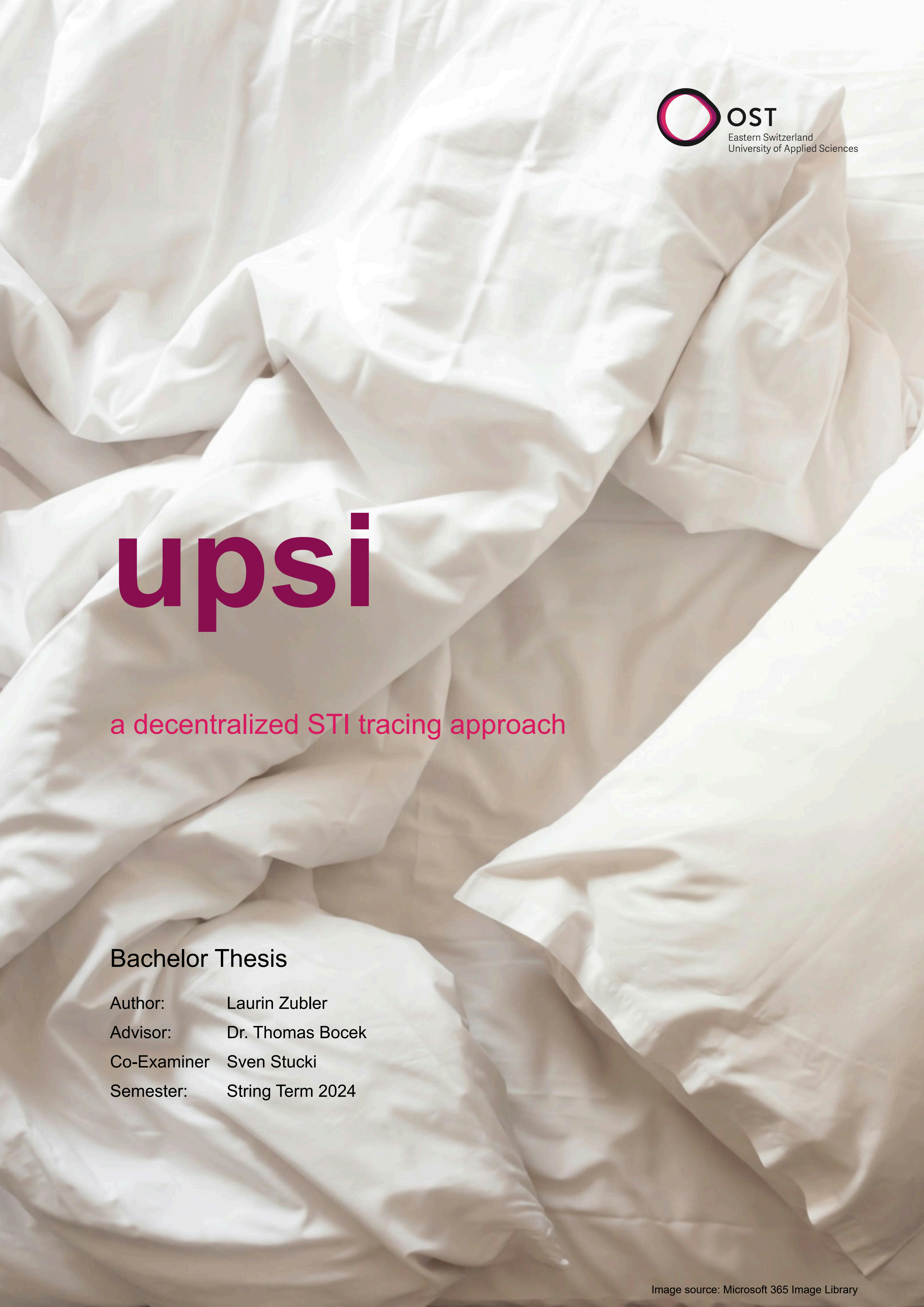


OST
Eastern Switzerland
University of Applied Sciences

upsi

a decentralized STI tracing approach

Bachelor Thesis

Author:	Laurin Zubler
Advisor:	Dr. Thomas Bocek
Co-Examiner	Sven Stucki
Semester:	String Term 2024

Image source: Microsoft 365 Image Library

Abstract

Introduction

Sexually Transmitted Infections (STIs) are a significant global public health challenge. While in Switzerland the incidence of Human Immunodeficiency Virus (HIV) has been declining since the 1980s pandemic, other STIs such as Chlamydia, Gonorrhea, and Syphilis exhibit an upward trend. Effective partner notification is essential to mitigate the spread of STIs. However, it is not practiced sufficiently, and no dedicated technical solution currently addresses this challenge.

During the COVID-19 pandemic, proximity tracing mobile apps were successfully deployed to combat the spread of SARS-CoV-2. Various system architectures were employed, utilizing different approaches concerning privacy and data sovereignty.

Objective

The primary objective of this thesis is to design and develop *upsi*, a mobile application for STI partner notification. Inspired by the COVID-19 proximity tracing apps, *upsi* aims to enhance partner notification, thereby mitigating the spread of STIs. Experts in the field of STIs will be consulted to evaluate the feasibility and importance of *upsi*.

Approach

Research was conducted to understand the current STI situation and existing solutions for STI partner notification and COVID-19 proximity contact tracing. A concept for *upsi* was developed based on insights gained from the research and presented to leading STI experts. The expert feedback was integrated into the solution design. A minimum viable product (MVP) was developed using the most feasible technologies evaluated.

Results

upsi, a partner notification application for STI rapid tests, was developed with a focus on privacy and decentralization. The solution consists of a Flutter mobile app for users, which provides contact exchange and partner notification, as well as a second mobile app for test center employees to ensure trustworthy notifications. A .NET Core server application deployed to Azure handles the publication of positive test results onto the Optimism blockchain and simplifies wallet handling for the test center employees.

STI experts responded positively to the proposed concept and provided helpful inputs and insights that were integrated into *upsi*. While technical solutions for partner notification are discussed among experts, integration into existing IT systems remains challenging due to the large number of test centers, each using its own IT solution.

Further Work

Further development of *upsi* is suggested, including the extension to iOS mobile devices and additional features to enhance user experience and functionality. Integration into existing STI test center IT systems should be carried out to also handle laboratory tests. Additionally, a study to evaluate the effectiveness and acceptance of *upsi* among users should be conducted.

Table of Contents

1 Introduction

Preamble

This chapter outlines the main motivation and the primary aims of this thesis.

1.2 Motivation

Introduction

In this section, key terms related to sexually transmitted infections (STIs) are described, and global and local perspectives are discussed. Partner notification and its challenges are explained, along with parallels to COVID-19 contact tracing efforts.

1.2.2 Sexually Transmitted Infections

Explanation of Terms [1]

Sexually Transmitted Infections (STI) are infections primarily spread through sexual activities. These include more than 30 different known bacteria, viruses, and parasites.

Among the eight STIs with the highest incidence worldwide, four are curable:

- Syphilis
- Gonorrhea
- Chlamydia
- Trichomoniasis

And four are incurable:

- Hepatitis B
- Herpes Simplex Virus (HSV)
- Human Immunodeficiency Virus (HIV)
- Human Papillomavirus (HPV)

Global Perspective [2]

According to the World Health Organization (WHO), STIs are major public health threats. Over 2.3 million deaths are caused by STIs annually, representing 14% of all deaths from infectious and parasitic diseases, digestive diseases, and cancer. 1.2 million new cancer cases are attributed to STIs each year. Daily there are more than 1 million new infections.

Switzerland [3]

According to the Swiss Federal Office of Public Health (FOPH), HIV diagnoses have been decreasing since 2002 due to the effectiveness of HIV prevention in Switzerland. However, a continuous upward trend in the number of cases is observed for Chlamydia, Gonorrhea, and Syphilis.

The downward trend of HIV cases in Switzerland is illustrated in Figure 1, while the upward trends of Chlamydia, Gonorrhea, and Syphilis are illustrated in Figure 2 to Figure 4. During the COVID-19 pandemic a dip in the case numbers can be seen in the diagrams.

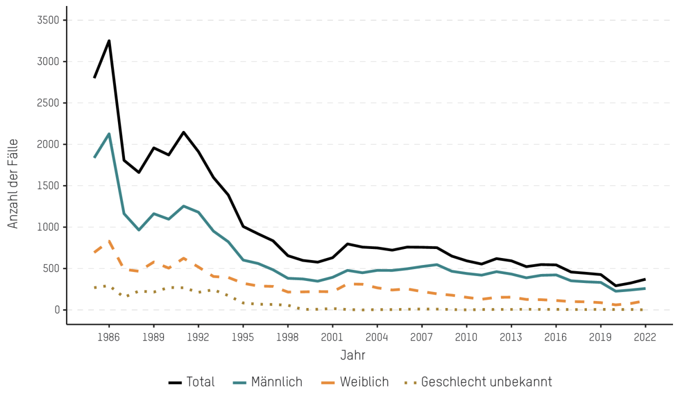


Figure 1: HIV laboratory reports by gender and year of testing since the start of testing, 1985–2022 [3]

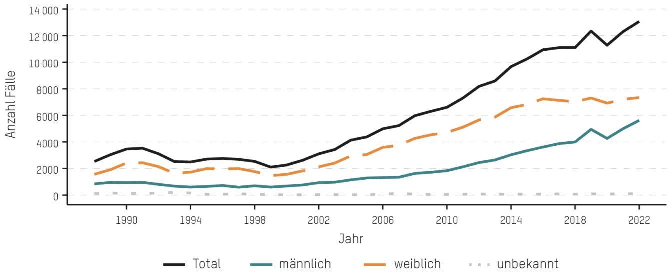


Figure 2: Chlamydia cases by gender and year of diagnosis since the start of recording, 1988–2022 [3]

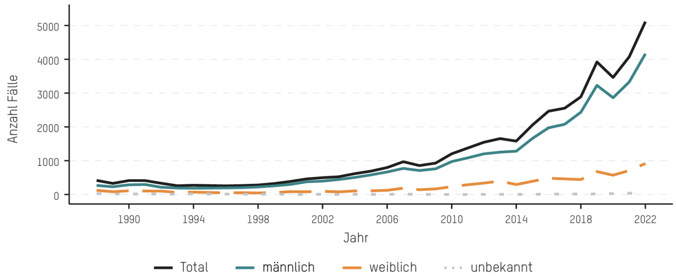


Figure 3: Gonorrhea cases by gender and year of diagnosis since the start of recording, 1988–2022 [3]

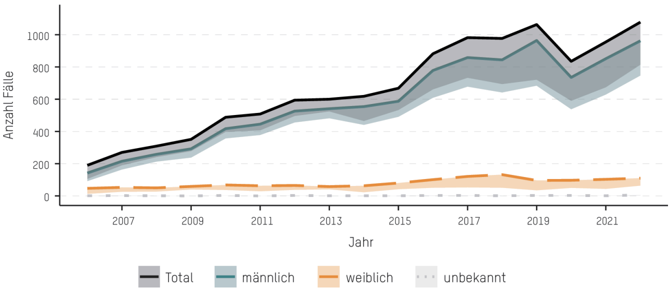


Figure 4: Syphilis cases by gender and year of diagnosis since the start of recording, 2006–2022 [3]

1.2.3 Partner Notification

Explanation of Terms [4]

Partner notification, also known as contact tracing, is the process by which individuals who have been in contact with an infected person are informed about their potential exposure.

There are multiple goals associated with partner notification:

- **For the infected person:** To eradicate the infection and prevent reinfection.
- **For contacts:** To identify and treat undiagnosed infections, thereby preventing further transmission.
- **For the population:** To interrupt transmission chains and reduce the overall spread of the infection.

STI Context [4]

Despite the importance of partner notification, many STI infected individuals are not contacting their sexual partners.

Several reasons are contributing to this behavior:

- **Fear of stigma and discrimination:** Individuals are afraid of judgement or ostracism by their peers and community.
- **Missing contact information:** Sexual partners are unknown or contact information is not available.
- **Emotional and physical abuse:** Emotional or even physical abuse is risked, particularly in relationships with a history of violence.
- **Lack of knowledge:** The importance of partner notification is not fully understood.
- **Privacy concerns:** The health status is considered confidential.
- **Relationship dynamics:** Fear of damaging the relationship or causing emotional pain.

COVID-19 Context

During the COVID-19 pandemic, mobile apps were used for proximity contact tracing by governments worldwide. Despite the different wording, proximity contact tracing involves the same process as STI partner notification. The SwissCovid app was effectively applied in Switzerland. Between 500 and 1000 SARS-CoV-2 positive cases per month were identified through the SwissCovid app. In 76% of the cases, preventive actions were taken by recipients after receiving an exposure report. [5]

Bluetooth was used to anonymously log encounters between users but concerns regarding privacy and security issues were raised by technical experts. The app core functionalities were controlled by Apple and Google, limiting transparency and flexibility and the risk of creating false alerts by malicious actors was expressed. Additionally, the contact tracing was only working in Switzerland as every country was using its own solution. [6]

Sensitive Data STI partner notification requires the processing of very sensitive data. Both health data and information about sexual partners are required. Additionally, the incidence of STIs among men who have sex with men (MSM) is higher than in the rest of the population. This group faces stigmatization in many countries and in some parts of the world even criminalization. A world map illustrating the global situation of laws regarding sexual orientation is presented in Figure 5.

The combination of these three factors requires strong privacy measures in a technical STI partner notification solution.

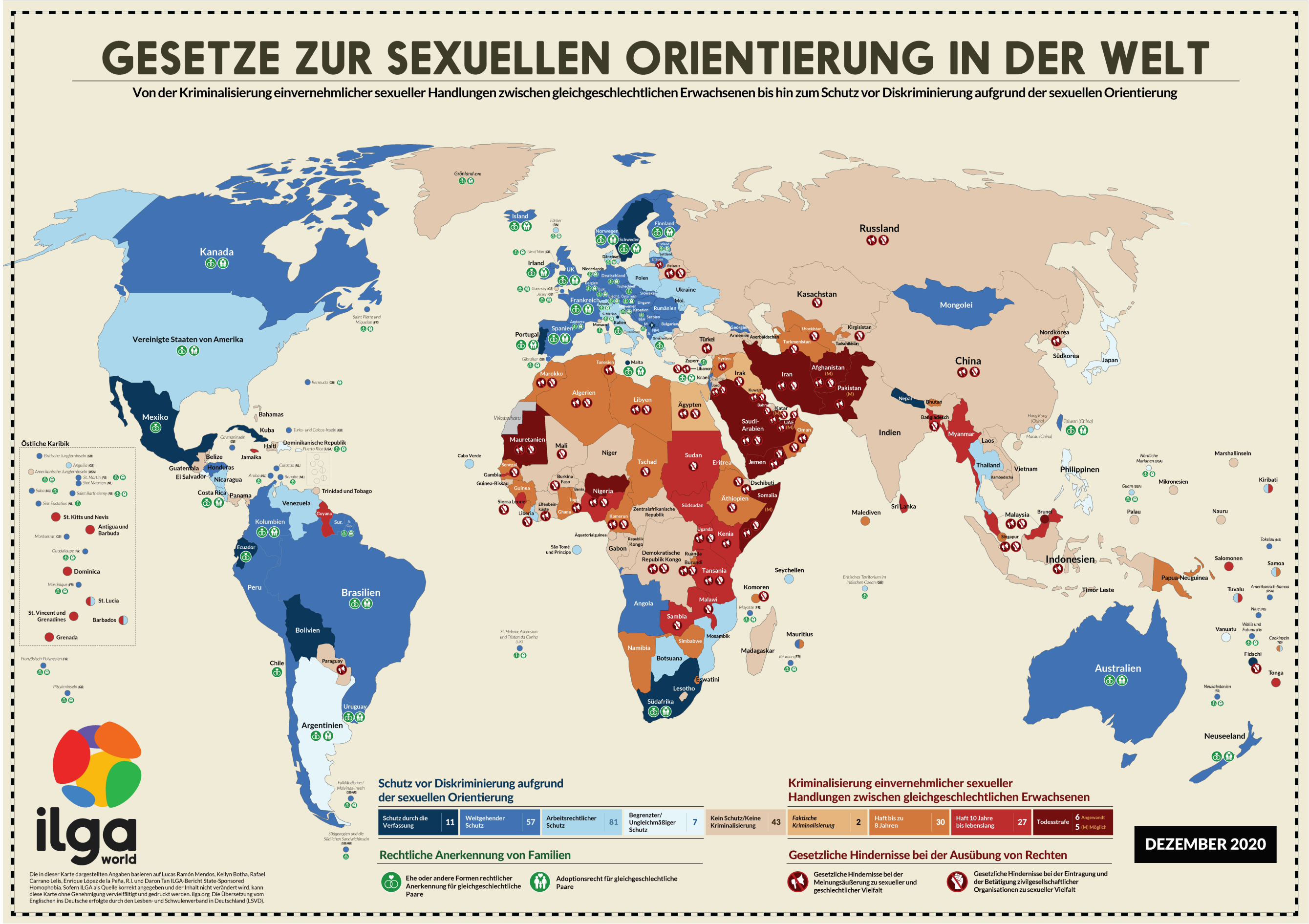

GESETZE ZUR SEXUELLEN ORIENTIERUNG IN DER WELT
Von der Kriminalisierung einvernehmlicher sexueller Handlungen zwischen gleichgeschlechtlichen Erwachsenen bis hin zum Schutz vor Diskriminierung aufgrund der sexuellen Orientierung
Kanada
Vereinigte Staaten von Amerika
Mexiko
Östliche Karibik
Kuba
Bahamas
Brasilien
Argentinien
Bolivien
Peru
Kolumbien
Venezuela
Chile
Uruguay
Paraguay
Russland
Kasachstan
Mongolei
China
Indien
Iran
Afghanistan
Pakistan
Saudi-Arabien
Jemen
Ägypten
Libyen
Algerien
Marokko
Mauretanien
Mali
Niger
Tschad
Sudan
Nigeria
Äthiopien
Somalia
Kenia
Tansania
Angola
Sambia
Namibia
Botsuana
Südafrika
Lesotho
Eswatini
Madagaskar
Indonesien
Philippinen
Japan
Australien
Neuseeland
Papua-Neuguinea
Türkei
Polen
Ukraine
Frankreich
Spanien
Portugal
Irland
Island
UK
Schutz vor Diskriminierung aufgrund der sexuellen Orientierung
Schutz durch die Verfassung 11
Weitgehender Schutz 57
Arbeitsrechtlicher Schutz 81
Begrenzter/ Ungleichmäßiger Schutz 7
Kein Schutz/Keine Kriminalisierung 43
Kriminalisierung einvernehmlicher sexueller Handlungen zwischen gleichgeschlechtlichen Erwachsenen
Faktische Kriminalisierung 2
Haft bis zu 8 Jahren 30
Haft 10 Jahre bis lebenslang 27
Todesstrafe 6 Angewandt 5 (M) Möglich
Rechtliche Anerkennung von Familien
Ehe oder andere Formen rechtlicher Anerkennung für gleichgeschlechtliche Paare
Adoptionsrecht für gleichgeschlechtliche Paare
Gesetzliche Hindernisse bei der Ausübung von Rechten
Gesetzliche Hindernisse bei der Meinungsäußerung zu sexueller und geschlechtlicher Vielfalt
Gesetzliche Hindernisse bei der Eintragung und der Betätigung zivilgesellschaftlicher Organisationen zu sexueller Vielfalt
DEZEMBER 2020
ilga world

Figure 5: World map showing the global legal status of sexual orientation[1]

[1] https://www.lsvd.de/de/ct/1245-LGBT-Rechte-weltweit

1.3 Aim of this Thesis

Introduction

In this section, the three primary objectives of this thesis are described.

Research

A Research should be conducted to understand the technical approaches used in COVID-19 tracing apps and to identify existing technical solutions for STI partner notification. The findings should be incorporated into the solution design.

STI Tracing App

The effectiveness of tracing apps, demonstrated during the COVID-19 pandemic, should be applied to reduce the increasing STI incidences. A mobile app should be developed that prioritizes privacy and data security while providing efficient partner notification. The barriers in STI partner notification should be addressed, and solutions should be found for the technical concerns raised about COVID-19 tracing apps.

Evaluation

An evaluation should be conducted by consulting experts in the field of STIs to assess the feasibility and significance of such an app. The current testing procedures should be understood to identify the necessary requirements and functionalities. Existing technical systems should be identified, and potential integration should be examined to ensure compatibility and ease of adoption. The feedback received should be incorporated into the development process, refining the app’s design and functionalities.

2 Background

Preamble This chapter describes the concepts and technologies underlying the developed solution.

2.2 Decentralized Systems

Explanation of Terms A decentralized system is a network architecture where control, decision-making, and data are distributed across multiple nodes rather than being concentrated in a single central entity. Nodes operate independently and collaborate to achieve a common goal, often leveraging technologies like blockchain.

Blockchain Blockchain is a decentralized and distributed transactional database technology. Unlike traditional centralized databases, blockchains are not relying on a single central authority. Instead, distributed ledgers are used where all transactions are recorded and accessible to all participants in the network. [7]

A comparison between a centralized and a distributed database is shown in Figure 6. Users with read-only access are represented by blue circles. Write transactions to the database are indicated with green arrows.
In the centralized database, only the central authority is allowed to write, and read requests can also be blocked. In the distributed database, all users are allowed to read. The miners, represented by green circles, are responsible for writing data.

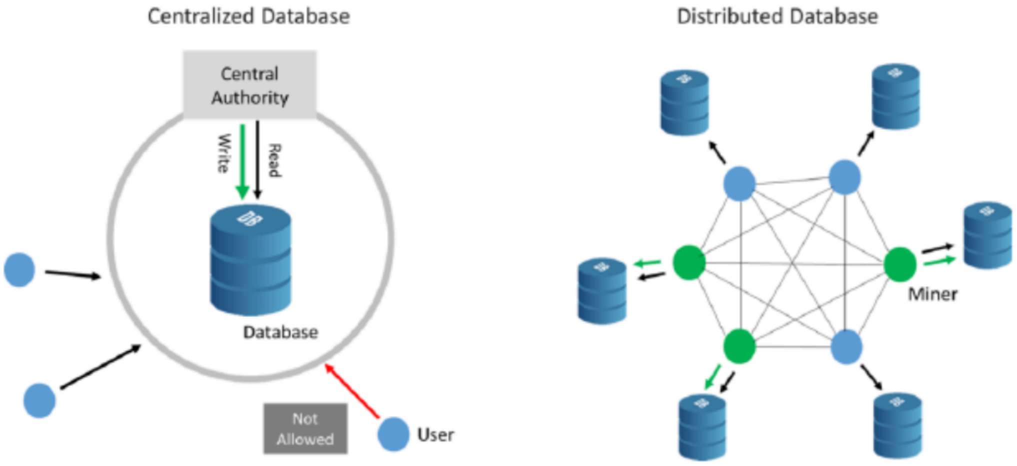


Figure 6: Centralized and distributed database comparison [8]

Block

Data in a blockchain is stored in blocks. The blocks are forming a chain by storing the hash of the previous block. This structure is ensuring the chain immutability, as altering a single block would require changing all following blocks. A simple chain containing three blocks is illustrated in Figure 7, demonstrating how each block references the hash of the previous block.

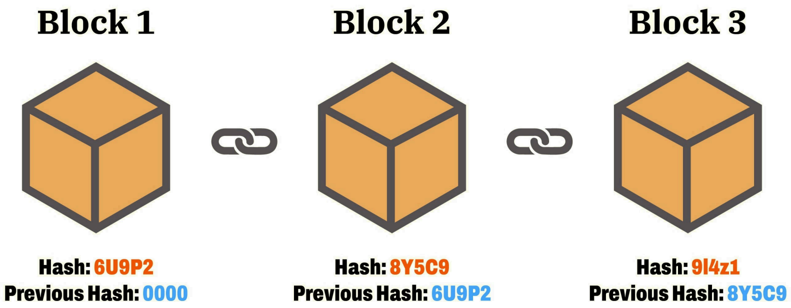


Figure 7: Simple blockchain with 3 blocks[2]

Blockchain Applications [9]

The first and most well-known blockchain application is Bitcoin[3], a decentralized payment network that is operating without the need for intermediaries such as banks. Bitcoin has led to the creation of numerous other cryptocurrencies, including Ethereum[4] and Tether[5].

Since then, Blockchain technology has been applied in a variety of sectors beyond cryptocurrencies:

- **Finance:** Digital assets, remittance, and online payments.
- **Healthcare:** Patient records management.
- **Supply chain management:** Transparency and traceability of goods from production to delivery.
- **Digital identity verification:** Identity records validated without a central authority.
- **Electronic voting:** Secure, transparent, and tamper-proof electoral processes.

Layer 2 Blockchain [10]

A Layer 2 (L2) is a separate blockchain built on top of a main blockchain (Layer 1). With the increasing popularity and usage of a blockchain, transaction fees rise. Layer 2 blockchain solutions are addressing these issues by processing transactions off-chain, which are then bundled and submitted to Layer 1 blockchain. This approach is providing faster transaction speed and lower fee costs.

DAO [11], [12]

A Decentralized Autonomous Organization (DAO) is a community-owned entity without centralized leadership, governed by blockchain-based rules that ensure transparency and democratized decision-making. DAOs are using smart contracts to enforce rules, automate decision-making processes and handle funds securely.

[2] https://money.com/what-is-blockchain/
[3] https://bitcoin.org/
[4] https://ethereum.org/
[5] https://tether.to/

Smart Contract [13]

A smart contract is executable code that is operating on the blockchain to enforce the terms of agreement between untrusted parties. It is ensured by the smart contract that all conditions are met and assets are released to the appropriate parties, eliminating the need for a trusted third party. Ethereum is the most common blockchain for smart contracts due to its programming language, *Solidity*[6].

Mining

Mining is the process by which multiple transactions are combined into a block and added to the blockchain.

Ther are two diffrent main consensus mechanisms used by blockchains for block creation:

- **Proof of Work (PoW):** Miners are required to solve a cryptographic puzzle. The first miner to solve the puzzle is allowed to add the next block to the blockchain and is rewarded for the effort. PoW is characterized by high energy consumption due to the significant computational power required.
- **Proof of Stake (PoS):** Validators are chosen to create new blocks based on the amount of cryptocurrency they hold and are willing to "stake" as collateral. Instead of solveing puzzles, validators are selected random manner. PoS is more energy-efficient compared to PoW.

Gas

Gas is a unit of measure used to calculate the transaction fees required to execute functions on a smart contract. The transaction fee is paid to the miner or validator of the block to reward their effort. The more actions are executed in the smart contract and the more data is saved on the blockchain, the higher the gas fee. The gas price depends on the network's current load.

[6] https://soliditylang.org/

2.3 QR Code

Explanation of Terms

The Quick Response (QR) code is a two-dimensional barcode used to encode information. QR codes are composed of black and white squares arranged in a grid and can be read by cameras.

An example of a QR Code is shown in Figure 8.

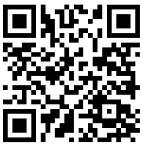


Figure 8: Example QR code

Applications

With the increasing popularity of smartphones and the greater availability of cameras and screens, the use of QR codes has also been more widespread. In Switzerland, QR codes are used by a variety of mobile apps to transfer data to a smartphone, either from another smartphone or from a static source.

The following applications are using a QR code:

- **SBB:** QR codes are used by the Swiss Federal Railways app to display train tickets or subscriptions (Figure 9).
- **Twint:** QR codes are utilized by a widely used mobile payment system to capture the recipient (Figure 10).
- **SwissCovid:** During the COVID-19 pandemic, QR codes were employed to present COVID certificates and capture event codes (i-covid).

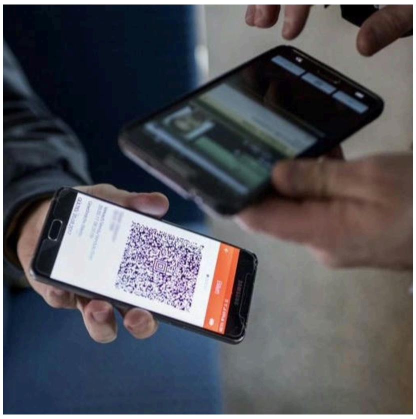


Figure 9: SBB mobile app[7]

Figure 10: Twint mobile app[8]

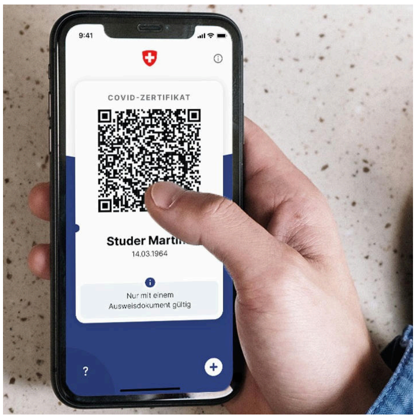


Figure 11: SwissCovid mobile app[9]

[7] https://www.rts.ch/info/suisse/11154401-les-suisses-se-sont-rues-sur-les-billets-degriffes-en-2019.html

[8] https://www.twint.ch/geschaeftskunden/unsere-loesungen/qr-code-sticker/

[9] https://www.bit.admin.ch/bit/de/home/themen/stories/covid-zertifikat.html

2.4 BLS digital signature

Explanation of Terms [14]

The Boneh-Lynn-Shacham (BLS) digital signature is a cryptographic signature scheme, providing the functionality to combine multiple signatures into a single short signature. This is useful when the same document needs to be signed by multiple parties or when signatures from multiple sources need to be combined efficiently.

Following steps are involed in creating an verifying a BLS signature:

1. **Signature Generation:** Multiple signatures are created on the same message using different private keys.
2. **Signature Aggregation:** The signatures are aggregated together, resulting in a single short aggregate signature.
3. **Verification:** The aggregate signature can be verified against the combined public keys of the signers and the message. This is ensuring that each message was signed by the respective private key holder.

The BLS process is visualized in Figure 12.

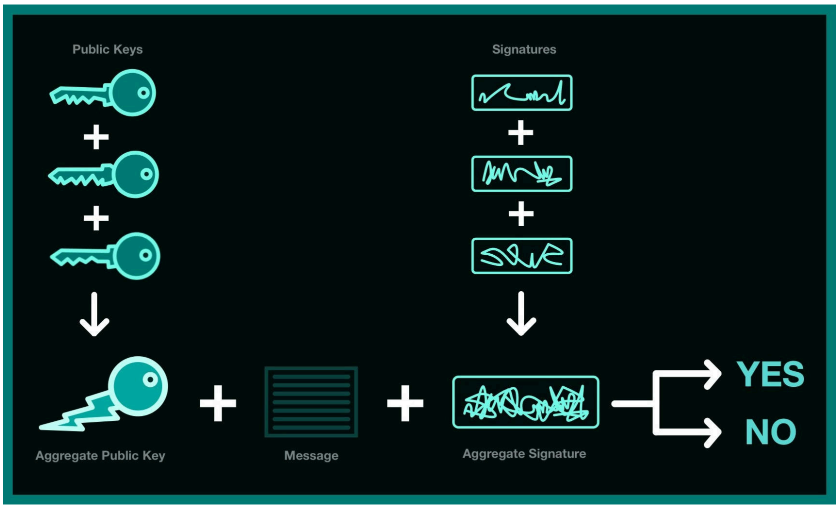


Figure 12: BLS signature functionality[10]

[10] https://inevitableeth.com/home/concepts/bls-signatures

2.5 STI Tests

Method

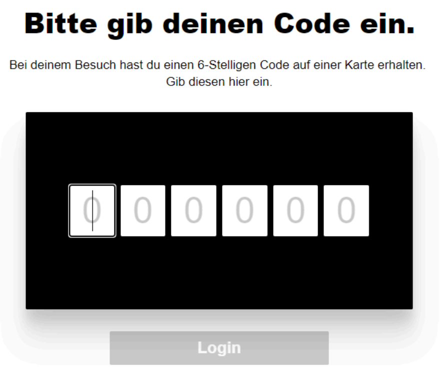
The testing procedure varies depending on which STI is being tested. Rapid tests are available for HIV, Syphilis, and Hepatitis C, providing results within 20 minutes. For other STIs, such as Chlamydia or Gonorrhea, a sample must be sent to a laboratory, and the results are available within two to five days.

Results

For rapid tests, the STI test result is communicated by the tester in person.

For laboratory tests, each test center has its own method of informing the individual about their results. These methods vary in their technical complexity. At some centers, such as Checkpoint[11], the result can be accessed online using a code.

Screenshots of the web Checkpoint web application are presentend in Figure 13 and Figure 14.

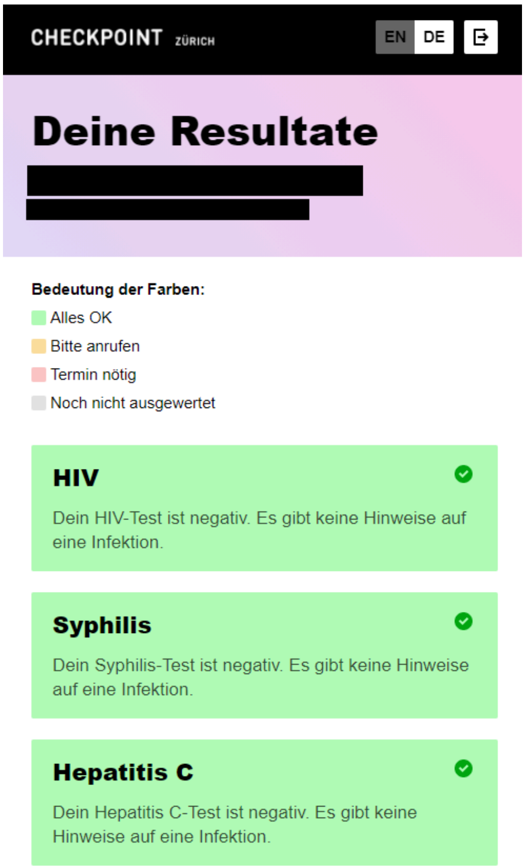


Bitte gib deinen Code ein.

Bei deinem Besuch hast du einen 6-Stelligen Code auf einer Karte erhalten. Gib diesen hier ein.

0 0 0 0 0 0

Figure 13: Screenshot of the Checkpoint STI test result web application code enter screen.[12]



Deine Resultate

[redacted]

Bedeutung der Farben:

- 🟩 Alles OK
- 🟧 Bitte anrufen
- 🟥 Termin nötig
- ⬜ Noch nicht ausgewertet

HIV ✓

Dein HIV-Test ist negativ. Es gibt keine Hinweise auf eine Infektion.

Syphilis ✓

Dein Syphilis-Test ist negativ. Es gibt keine Hinweise auf eine Infektion.

Hepatitis C ✓

Dein Hepatitis C-Test ist negativ. Es gibt keine Hinweise auf eine Infektion.

Figure 14: Screenshot of the Checkpoint STI test result web application result page.[13]

[11] https://www.cpzh.ch/

[12] https://www.cpzh.ch/angebote/online-resultate/

[13] https://www.cpzh.ch/angebote/online-resultate/

3 Related Work

Preamble This chapter describes and discusses the related techincal solutions identified.

3.2 STI Related Mobile Apps

Introduction In this section, various mobile apps in the STI context are furhter detailed.

mHealth App [15] To prevent HIV transmission among men who have sex with men (MSM), a study was conducted to explore the effectiveness of a mobile health (mHealth) app. Partner notification is the core feature of the app, designed to help MSM communicate their HIV status before meeting. The users health status is stored centrally in the health center and can be requested by other users. The HIV test results were uploaded to the health center by the health center staff.

The partner notification process is further detailed in Figure 15 and screenshots of the app are presented in Figure 16

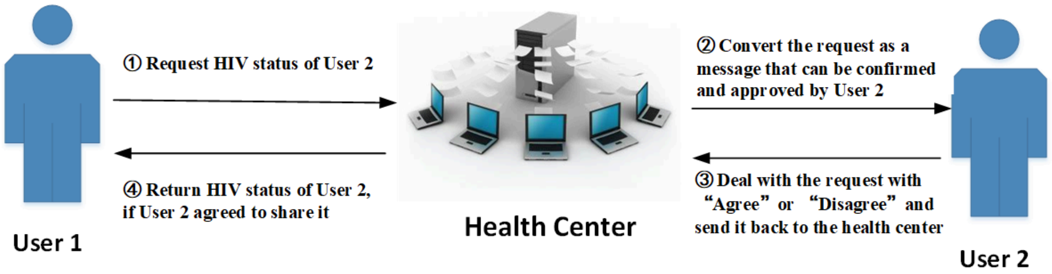
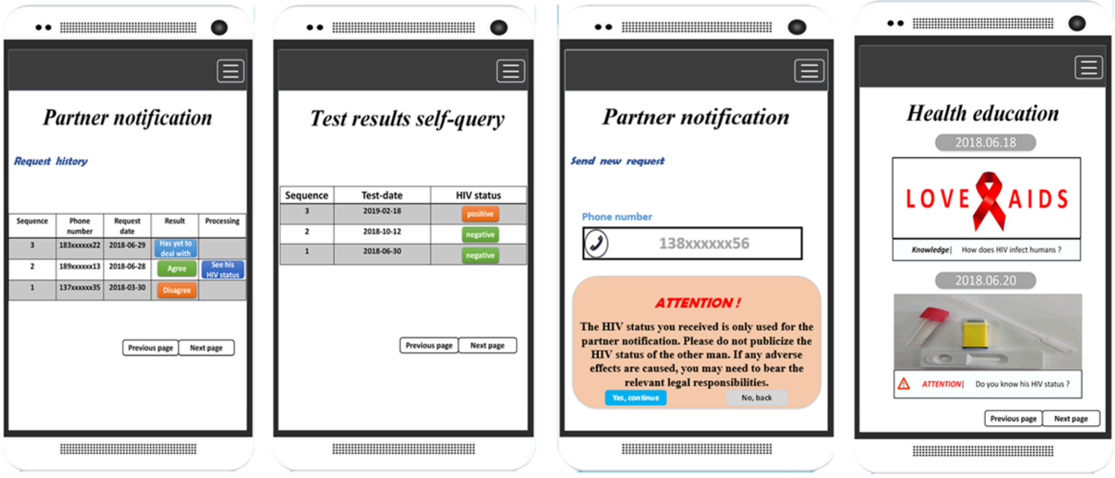

User 1
① Request HIV status of User 2
④ Return HIV status of User 2, if User 2 agreed to share it
Health Center
② Convert the request as a message that can be confirmed and approved by User 2
③ Deal with the request with "Agree" or "Disagree" and send it back to the health center
User 2

Figure 15: mHealth app partner notification process [15]



Partner notification

Request history

Sequence	Phone number	Request date	Result	Processing
3	183xxxxxx22	2018-06-29	Has yet to deal with	
2	189xxxxxx13	2018-06-28	Agree	See his HIV status
1	137xxxxxx35	2018-03-30	Disagree	

(a) Partner notification module

Test results self-query

Sequence	Test-date	HIV status
3	2019-02-18	positive
2	2018-10-12	negative
1	2018-06-30	negative

(b) Test results self-query module

Partner notification

Send new request

Phone number: 138xxxxxx56

ATTENTION !

The HIV status you received is only used for the partner notification. Please do not publicize the HIV status of the other man. If any adverse effects are caused, you may need to bear the relevant legal responsibilities.

(c) Prompt and warning module

Health education

2018.06.18

Knowledge| How does HIV infect humans ?

2018.06.20

ATTENTION| Do you know his HIV status ?

(d) Health education module

Figure 16: mHealth app screenshots [15]

mHealth App Results [15]

The study resulted that users who used the app for more than 5 months showed a significantly lower HIV incidence (2.22 per 100 person-years) compared to others (6.99 per 100 person-years).

The centraliced data collection also allowed a detailed analysis of the users location and sexual contacts as shown in Figure 17 and Figure 18.

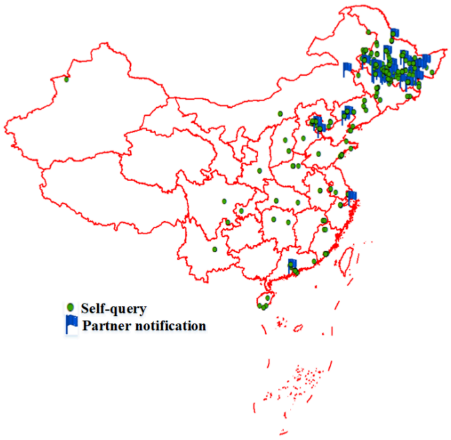


Figure 17: Map of China and some independend island states showing the local distribution of the mHealth app usage [15]

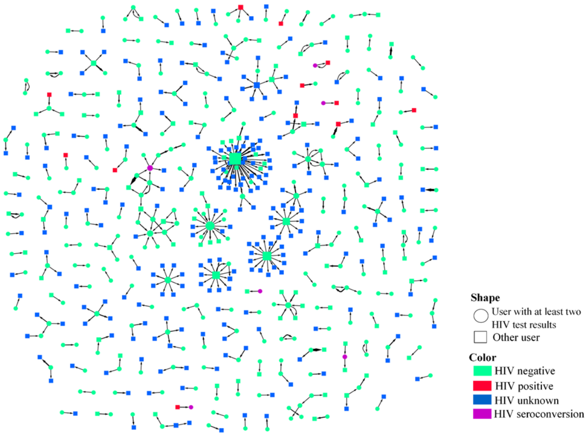


Figure 18: Contact networks formed by the mHealth app partner notification [15]

LYNX App

To promote HIV/STI testing and the uptake of PrEP among young men who have sex with men (YMSM) a study was developing and testing a mobile app (LYNX). Among other features, an electronic diary to track sexual encounters was included in the app. No detailed description of the diary was found; it is assumed that the diary is manually filled out by the user and could be used for manual partner notification in the event of an STI infection, although this was not the primary purpose of the diary. [16]

The LYNX app was found to be both feasible and acceptable among YMSM. Participants responded positively to the app's features, such as the electronic diary. [17]

DOT Diary App

To monitoring and support PrEP use among YMSM a study was developing and testing a mobile app (DOT Diary). Among other features, the app was including a diary component, adapted from the LYNX mobile app. Participants were instructed to enter nicknames of sex partners to maintain confidentiality and partner specific information was encrypted and not transmitted to the study team to further protect privacy.

The app was found to be highly acceptable with high levels of app use. [18]

3.3 Electronic Partner Notification Services

Introduction

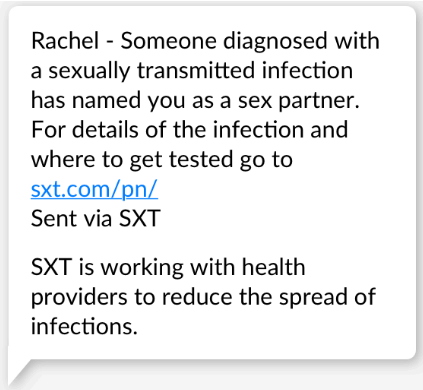
In this section, electronic services that assist sending partner notifications are detailed further.

SXT

In the United Kingdom, *SXT*, a community interest company, is offering a software solution for sexual health clinics to send anonymous partner notifications. Former sexual partners of an infected person are recorded with the help of a health employee. A notification via SMS or email is sent to the partners by *SXT*, including a code, which allows any clinic to check the originally diagnosed STI on the website of *STX*.

The whole process is further described at https://sxt.health/uk/pn/about.

A sample partner notification text message by SXT is displayed in Figure 19.



Rachel - Someone diagnosed with a sexually transmitted infection has named you as a sex partner. For details of the infection and where to get tested go to sxt.com/pn/
Sent via SXT

SXT is working with health providers to reduce the spread of infections.

Figure 19: SXT sample partner notification text message[14]

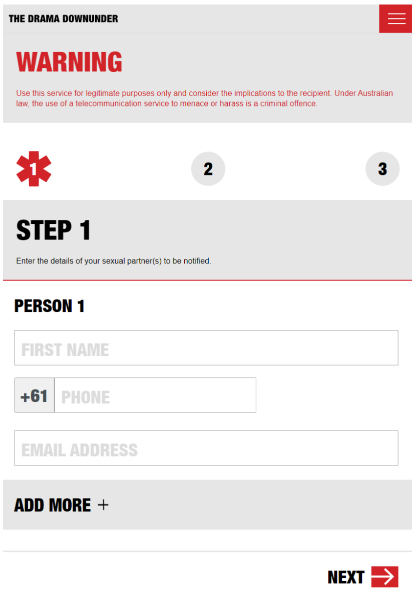
[14] https://sxt.health/uk/pn/about

The Drama Downunder

In Australia, partner notifications can be sent via SMS and email through a service of *The Drama Downunder*. The option to send messages anonymously is provided, and no authentication or identification is necessary.

The service is available at https://www.thedramadownunder.info/let-them-know/.

Screenshots from the parnter notification steps from *The Drama Downunder* are displayed in Figure 20, Figure 21, Figure 22. A parnter notification mail is presented in Figure 23.


THE DRAMA DOWNUNDER

WARNING

Use this service for legitimate purposes only and consider the implications to the recipient. Under Australian law, the use of a telecommunication service to menace or harass is a criminal offence.

STEP 1

Enter the details of your sexual partner(s) to be notified.

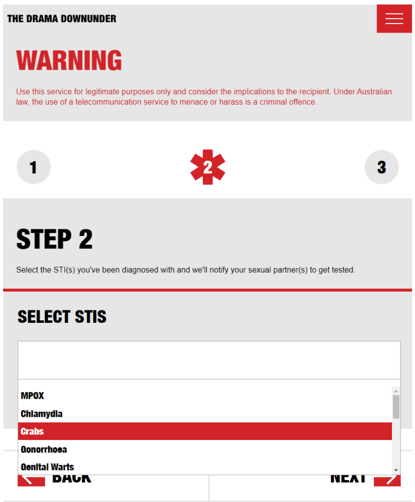
PERSON 1

FIRST NAME

+61 PHONE

EMAIL ADDRESS

Figure 20: Screenshot of The Drama Downunder partner notification step 1[15]


THE DRAMA DOWNUNDER

WARNING

Use this service for legitimate purposes only and consider the implications to the recipient. Under Australian law, the use of a telecommunication service to menace or harass is a criminal offence.

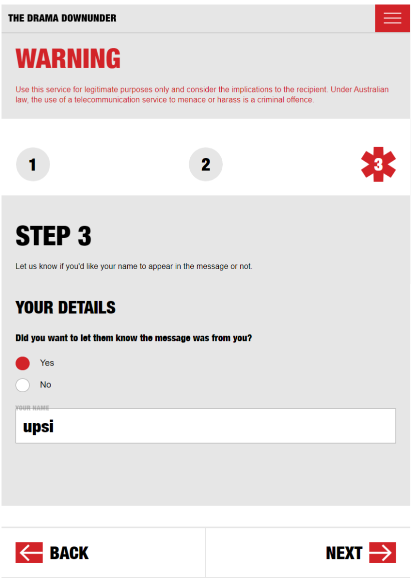
STEP 2

Select the STI(s) you've been diagnosed with and we'll notify your sexual partner(s) to get tested.

SELECT STIS

MPOX
Chlamydia
Crabs
Gonorrhoea
Genital Warts

Figure 21: Screenshot of The Drama Downunder partner notification step 2[16]


THE DRAMA DOWNUNDER

WARNING

Use this service for legitimate purposes only and consider the implications to the recipient. Under Australian law, the use of a telecommunication service to menace or harass is a criminal offence.

STEP 3

Let us know if you'd like your name to appear in the message or not.

YOUR DETAILS

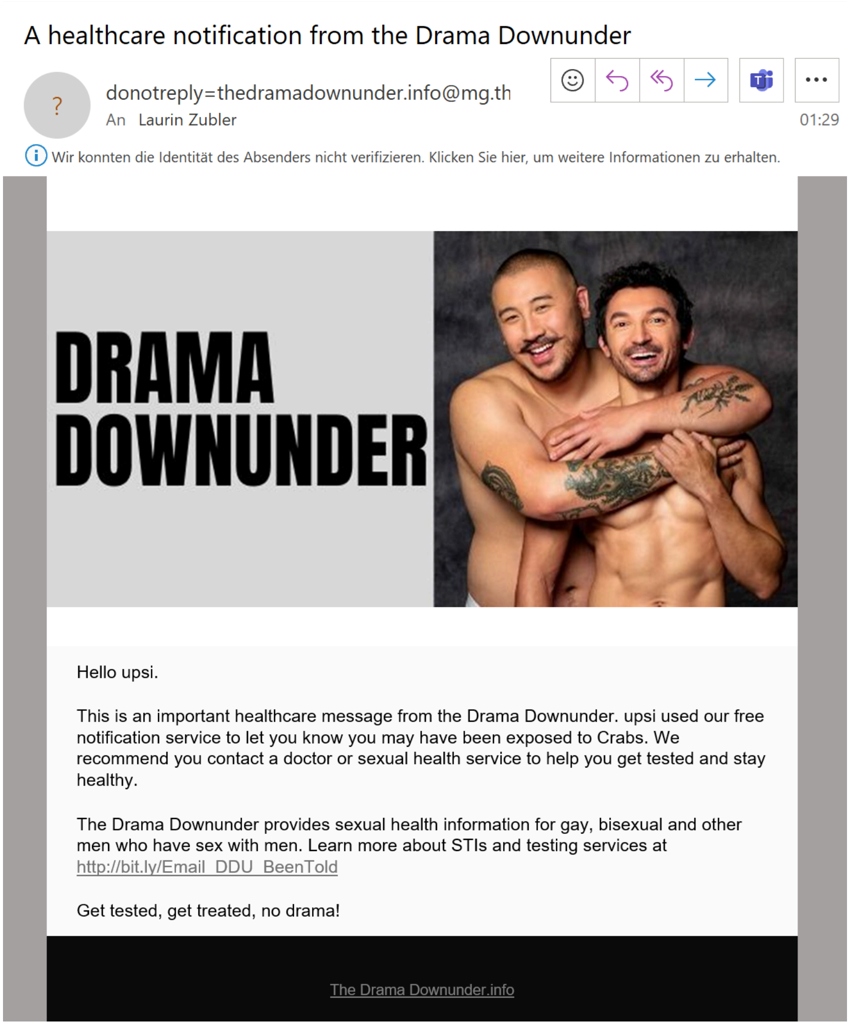
Did you want to let them know the message was from you?

● Yes
○ No

YOUR NAME
upsi

Figure 22: Screenshot of The Drama Downunder partner notification step 3[17]

[15] https://www.thedramadownunder.info/let-them-know/
[16] https://www.thedramadownunder.info/let-them-know/
[17] https://www.thedramadownunder.info/let-them-know/



A healthcare notification from the Drama Downunder

donotreply=thedramadownunder.info@mg.th
An Laurin Zubler
01:29

Wir konnten die Identität des Absenders nicht verifizieren. Klicken Sie hier, um weitere Informationen zu erhalten.

Hello upsi.

This is an important healthcare message from the Drama Downunder. upsi used our free notification service to let you know you may have been exposed to Crabs. We recommend you contact a doctor or sexual health service to help you get tested and stay healthy.

The Drama Downunder provides sexual health information for gay, bisexual and other men who have sex with men. Learn more about STIs and testing services at http://bit.ly/Email_DDU_BeenTold

Get tested, get treated, no drama!

The Drama Downunder.info

Figure 23: Screenshot of a The Drama Downunder partner notification mail

inSPOT

In the USA, *inSPOT*, a similar service to *The Drama Downunder* in Australia, was active. [19]

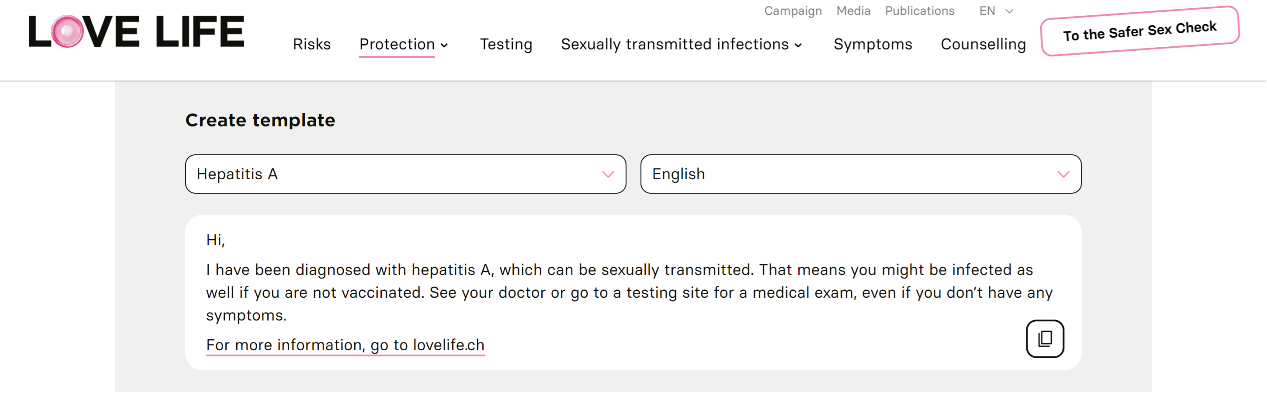
However, the service is no longer available at the specified URL, and no information is found about what happened.

Love Life

In Switzerland, pre-written texts for composing partner notifications are provided by *Love Life*.

The service is available at https://lovelife.ch/en/protection/communicating-in-sexual-relationships.

A screenshot of the partner notification template creator of *Love Life* is displayed in Figure 24.



Create template

Hepatitis A

English

Hi,

I have been diagnosed with hepatitis A, which can be sexually transmitted. That means you might be infected as well if you are not vaccinated. See your doctor or go to a testing site for a medical exam, even if you don't have any symptoms.

For more information, go to lovelife.ch

Figure 24: Screenshot of the Love Life partner notification template creator[18]

[18] https://lovelife.ch/en/protection/communicating-in-sexual-relationships

3.4 Electronic Partner Notification Acceptance

Introduction

In this section, studies evaluating the acceptance of technical solutions for partner notifications are further detailed.

Systematic Review [20]

The acceptability and utilization of electronic communication technologies for STI partner notification are reviewed in a systematic review of 23 studies.

The key findings are:

- High levels of interest and acceptability were reported for electronic partner notification methods, such as anonymous e-cards, text messages, emails, and phone calls.
- Despite the interest, actual usage of these methods was relatively low.
- Electronic communications were found to be particularly useful for notifying less committed partners who might otherwise remain uninformed.

Survey [21]

An online survey was conducted among users of geosocial networking apps to assess the acceptability and preferences for app-based partner notification.

Three notification strategies were investigated:

1. Personal notification using a user profile.
2. Anonymous notification via a health department.
3. Anonymous in-app notification.

It was revealed by the study that app-based partner notification would be used and is acceptable.

Receiving partner notification

- 70% preferred to be notified directly by their partner.
- 95% would get tested if notified by a health department profile.
- 85% would get tested if notified via anonymous in-app message.

Sending partner notification

- 50% preferred notifying a partner using their own profile.
- 26% preferred health department assistance for notification.
- 24% preferred anonymous in-app messaging.

Comfort with notification methods

- 71% were comfortable with health department profiles being used for notifications.
- 74% were comfortable with anonymous in-app messaging.

3.5 COVID-19 Proximity Tracing

Introduction [22]

During the COVID-19 pandemic, mobile proximity tracing apps were developed to combat the spread of the SARS-CoV-2 virus and offering opportunities to ease lockdown measures. While notable success was observed in Asian countries to reduce infections using mobile tracing apps, these apps were highly invasive regarding user privacy and would not have been accepted by European citizens. Consequently, various architectures for privacy preserving proximity tracing apps were proposed across Europe, including several variants of DP-3T, the PEPP-PT framework, and the French ROBERT system.

DP-3T

DP-3T (Decentralized Privacy-Preserving Proximity Tracing) is a contact tracing protocol, developed by an international consortium of technologists, legal experts, engineers and epidemiologists. The term “decentralized” is refering to the decentralization of data storage and processing, as the contact information are stored locally on the device. A central server is used to alert other users in case of an infection.

More information at: https://github.com/DP-3T/

PEPP-PT and ROBERT

PEPP-PT (Pan-European Privacy-Preserving Proximity Tracing) and ROBERT (ROBust and privacy-presERving proximity Tracing) are two centralized contact tracing protocols. Both are using a central server for contact information data storage and management.

More information at: https://github.com/pepp-pt and https://github.com/ROBERT-proximity-tracing/.

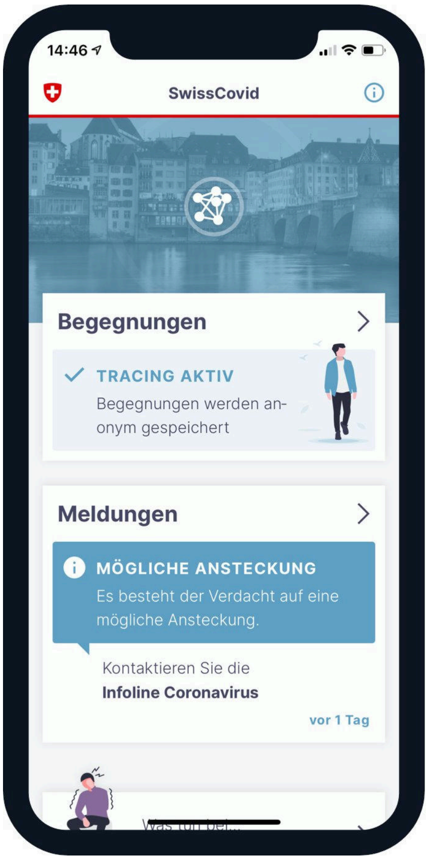
3.5.2 SwissCovid

Introduction

SwissCovid was the official contact tracing app of Switzerland and part of the Swiss Proximity Tracing System. On 1. September 2023, the Swiss Proximity Tracing System ceased operations.

The SwissCovid app is based on the DP-3T protocol. The source code, along with other information, can be viewed on the GitHub pages of SwissCovid[19] and Swiss Admin[20].

A screenshot of the SwissCovid app is displayed in Figure 25



Begegnungen werden anonym gespeichert

Es besteht der Verdacht auf eine mögliche Ansteckung.

Kontaktieren Sie die **Infoline Coronavirus**

Figure 25: Screenshot of the SwissCovid app[21]

[19] https://github.com/SwissCovid

[20] https://github.com/admin-ch/

[21] https://x.com/BAG_OFSP_UFSP/status/1265979020585435137

Components [23] The Swiss Proximity Tracing System consists of several frontend and backend components:

- **SwissCovid app:** The apps running on users iOS or Android devices.
- **DP3T SDK Backend:** The backend for uploading/providing the list of TEKs for proximity tracing.
- **CrowdNotifier Backend:** The backend for uploading/providing the list of exposed events for presence tracing.
- **HealthAuthority Backend:** The backend for issuing and validating Covidcodes.
- **CovidCode UI:** The web frontend for generating Covidcodes (used by health authorities).
- **Config Backend:** Backend component that serves configuration values for the SwissCovid app.
- **Additional Info Backend:** Backend component that serves the data for the statistics tab in the SwissCovid app.

An overview of the interaction between the different components is displayed in Figure 26.

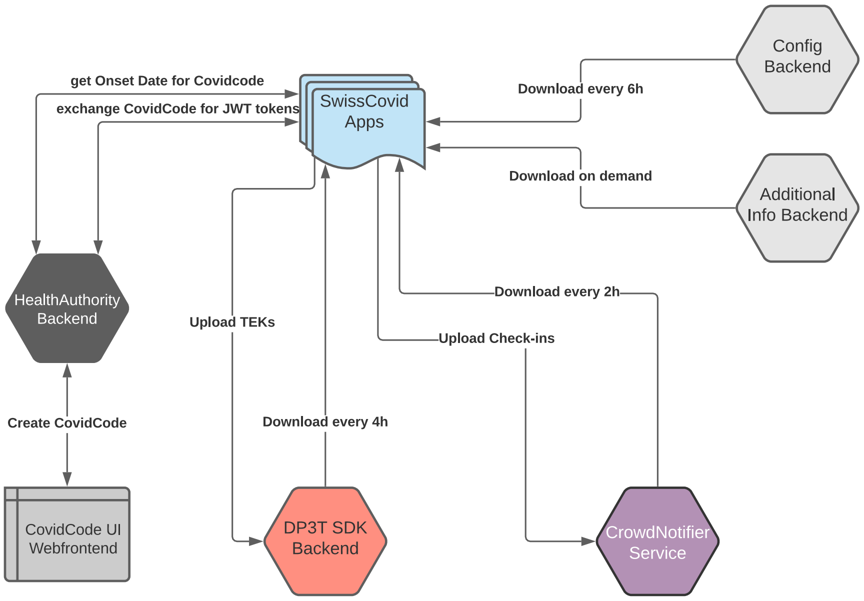


Figure 26: Swisscovid Architecture Overview [23]

App Functionality

The app is using two keys to prevent tracking:

- **Temporary Exposure Keys (TEKs):** Generated daily on the users device.
- **Rolling Proximity Identifiers (RPIs):** Derived from TEKs and renewed approximately every 10-20 minutes.

When two devices are coming into close proximity, RPIs are exchanged over Bluetooth and stored locally.

In case of a positive COVID-19 test, a 12-digit covidcode is received from the health authority. This code is entered into the app by the user and upon verification, the user’s recent TEKs are uploaded to the central backend server.

Devices periodically are downloading the list of TEKs from the server and checking locally if any are matching the known RPIs.

3.6 Discussion

Acceptance

The acceptance of STI-related mobile apps in a sex-positive community was positively evaluated in two studies. This can be seen as a good indication of the potential acceptance of *upsi*.

mHealth App

The mHealth app is raising some concerns regarding privacy and its effectiveness.

Although the study has been demonstrated that fewer HIV cases are observed among the mHealth app users, the results should be interpreted with caution, as participation is not incentivized for users who are HIV positive.

Despite the study referring to the UN partner notification definition, it is questionable whether the app implementation aligns with this definition. There is no notification of former sexual partners following a positive STI test.

Central data collection of sensitive health data as well as information on sexual contacts is required for the mHealth app. This is raising significant privacy concerns, as the data could be misused by an untrustworthy authority.

STX

The STX solution offers anonymous notification and standardized procedures that support healthcare staff. According to STX, no data is stored. It is questionable why the diagnosed STI is not included directly in the notification message. Instead, the code must be explicitly entered on the *SXT* website to learn the STI status. It is assumed that this was done to increase data collection and to raise the popularity of *STX* among clinics.

Privacy

The promised anonymity of the systems for partner notification is refering to the recipient not knowing who the positive-tested person is. Information about the sexual activity of the infected individual, as well as the mobile number or email address of their partners, is obtained by the services.

Misuse

The Drama Downunder allows partner notifications to be sent without an actual STI infection. An email can be triggered, as seen in Figure 23. This feature can be misused to send false partner notifications, causing unnecessary distress to individuals.

DP-3T

The SwissCovid app and the DP-3T protocol provide a well-researched and well-thought-out solution for partner notification that can be used as a foundation for the development of *upsi*. There are some points that could be improved, such as the centrally controlled database for publishing infections, which could be managed more decentralized.

Comparison The various related techincal solutions and *upsi* are compared in Table 1

	mHealth App	LYNX App	DOT Diary App	The Drama Downunder	inSPOT	Love Life	Swiss Covid App	upsi
Ensures privacy of the infected individual	✗	✗	✗	✓	✓	✓	✓	✓
Ensures privacy of the notified partner	✗	✗	✗	✗	✗	✓	✓	✓
No centralized data sovereignty	✗	✗	✗	✗	✗	✓	✗	✓
Provides STI partner notification	✗	✗	✗	✓	✓	✗	✗	✓
Prevents false partner notifications	✓	✗	✗	✗	✗	✗	✓	✓
Still in service	✗	✗	✗	✓	✗	✓	✗	✓
Open source code base	✗	✗	✗	✗	✗	✗	✓	✓

Table 1: Comparison of related techincal solutions and upsi

4 Solution Design

Preamble

This chapter outlines the solution design for the *upsi* application, detailing the methodology and system architecture.

4.2 Methodology

Introduction

In this section, the various methods and approaches utilized during the bachelor's thesis are furhte detailed.

Introduction

This bachelor's thesis is the continuation of a project developed in the blockchain course of fall 2023. The idea of an STI tracing app was considered exciting and was chosen to be pursued further with more time.

Research

A research was undertaken to understand the current STI situation and technical solutions in the STI context. The STI partner notification and Covid-19 proximity tracing were examined, with a specific focus on the technical implementation of the SwissCovid app. The results are presentend in chapter 1 and chapter 3.

Concept

A concept for *upsi*, an STI partner notification application was developed based on insights gained from the research phase and an interactive mobile app design prototype was created using Figma[22]. Improvements to the concept and design prototype were made continuously, incorporating the feedback from STI experts.

The results are presented in chapter 4.3.

Evaluation

Feedback was sought from leading STI experts in Switzerland, including researchers in STI prevention and treatment, STI consultation and education professionals, employees at STI test centers, and IT software company representatives who have developed applications in the STI field.

The primary goals were to assess whether *upsi* was a viable idea, evaluate and improve the concept, understand STI testing procedures, and determine how the app could be integrated into existing procedures and IT systems.

The expert feedback is presentend in chapter 6.2.

Technology Evaluation

Various mobile application and blockchain technologies were evaluated to determine the most suitable components for *upsi*. For the mobile app, platform approaches were evaluated, feature support was considered and developer-friendliness was assessed. Blockchains were evaluated based on transaction price and speed, trustworthiness and developer-friendliness. Server frameworks were evaluated for security, feature support and developer-friendliness.

The results are presented in chapter 5.

[22] https://www.figma.com/

Implementation

A minimum viable product (MVP) was implemented that incorporated the core functionalities of the system using the evaluated technologies. The MVP development was based on the research findings, design prototype, and industry feedback. Unit tests were conducted to ensure functionality. The server components were deployed to Azure.

The solution is further detailed in chapter 5.

4.3 System Architecture

Introduction In this section the *upsi* system requirements, challenges and overview are furhter detailed.

4.3.2 System Requirements

Decentralized Following the DP-3T approach, contact data should be stored locally on the user's device. This prevents the misuse of sensitive data.

Anti Authoritarian No single organization should have control over the infection event data. This ensures the longevity of the system and allows other parties to join. By distributing control, the integrity and trustworthiness of the system are maintained.

Privacy No user data should be stored. Only in the event of an infection is the necessary data published.

User Friendly The use of the app should not require technical knowledge and should be free of charge. This eliminates entry barriers, ensuring that as many people as possible use the app.

Trustworthy Partner Notification Only verified test results should allow for sending partner notifications. This guarantees the trustworthiness of the system.

Open The source code should be fully open-source, allowing for transparency and community trust.

Consent Users must always provide consent before any action is taken. No operations are performed without the user's knowledge, ensuring transparency and trust in the system's processes.

4.3.3 Challenges

Blockchain A blockchain is necessary to accomplish decentralized data storage. Blockchain usage requires technical knowledge, handling wallets and cryptocurrencies. Additionally, saving data to a blockchain incurs transaction fees. This contradicts the requirements and cannot be expected of the users.

Trust To ensure trustworthy partner notification, some identity in the system must be trusted. This cannot be the user, as no user data is stored for privacy reasons, and therefore no login is possible.

4.3.4 Overview

System Architecture

The system architecture for *upsi* is based on the DP-3T protocol used for the SwissCovid app.

Some deviations have been made to the SwissCovid architecture:

- **Blockchain:** Instead of a central database, a blockchain is used for storing infection events, to prevent data sovereignty by a single actor.
- **Keys:** No RPIs are used, the public keys are exchanged directly. This is possible because the public keys are not constantly broadcasted via Bluetooth. Additionally, the keys are valid for a longer period, assuming that fewer contacts and infections occur.
    - **Partner Notification:** In DP-3T, upon a positive test, the infected person receives a code from the health authority to trigger partner notification.

  For positive STI tests, no written document is issued, as these often take place anonymously. To prevent false partner notifications, only test center employees are allowed to trigger them, and the user's test presence is verified by signing a proof of attendance.

The *upsi* system overview is visualized in Figure 27 and further explained on the next page.

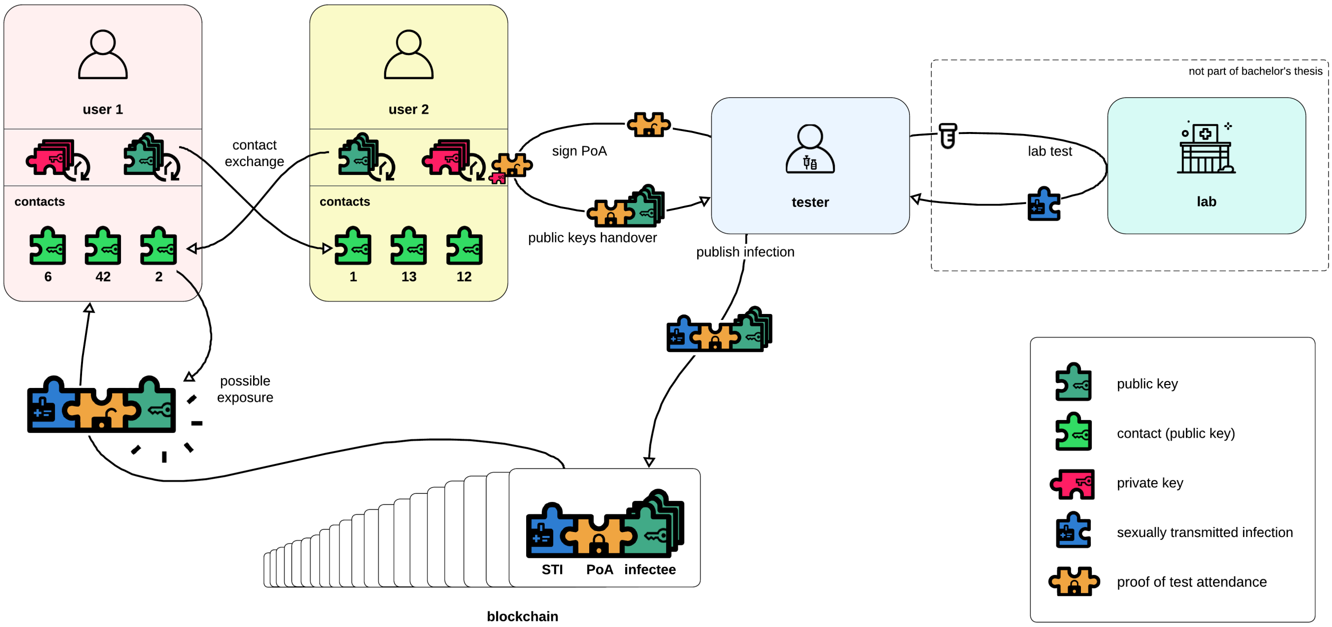


Figure 27: upsi *system overview*

4.3.4.2 Actors

User At a sexual contact, users exchange their public keys with other users. This interaction is labeled with *contact exchange* in Figure 27. Users are notified if someone they have been in contact with is tested positive, labeled with *possible exposure* in Figure 27.

Tester Testers are the trusted entities in the system. Testers conduct STI tests and publish infections to the blockchain. They are also responsible for paying the associated gas fees.

Lab Laboratory tests require sending test samples to the laboratory for analysis. The integration of laboratories is not part of this bachelor's thesis.

4.3.4.3 Test

Procedure If a user is tested positive for an STI, the users public keys are handed over to the tester, and the test attendance is verivied by signing the Proof of Attendance (PoA). This process is labeled as *sign PoA* and *public keys handover* in Figure 27.

4.3.4.4 Key Handling

Private / Public Keys Users possess private and public keys. The private key is used to verify the test, ensuring that the data written to the blockchain by the tester is accurate.

Regeneration Keys are regenerated at regular intervals to enhance privacy. This approach ensures that when a key is published, the entire STI history of a person is not revealed, thus protecting user privacy over time.

4.3.4.5 Blockchain

Pubilsh Infections STI infections are saved to the blockchain by the tester. Since all data on a blockchain is publicly accessible, the STI incident is thereby published. Users can read the infections on the blockchain and check if they have been in contact with the infected person.

Why Blockchain? The *upsi* system would also work with a centralized database to save the infection data. However, the blockchain offers several advantages over a traditional database.

- **Anti Authoritarian:** The use of blockchain is eliminating the need for a central authority, reducing the risk of data misuse and enhancing trust among users. Without a single entity controlling the data, transparency and security are ensured.
- **Open Access:**Open access is provided by blockchain technology, meaning the system is not restricted to a specific country or company. Other systems can be built using the same infection data on the blockchain, enabling a global partner notification system.
- **Durability:** The blockchain is still functioning and guaranteeing the access to the data even if *upsi* is out of service.
- **Privacy by Design:** As all data is publicly visible on the blockchain, the system is designed to operate without storing any sensitive information. This approach is preventing user data misuse and the occurrence of data leaks.

5 Implementation

Preamble

This chapter describes the implementation details of the *upsi* application. An overview of the system architecture is provided, and each component is detailed further. The limitations encountered during the implementation are discussed at the end.

5.2 System Components

Introduction

The *upsi* system consists of three main components: The mobile app, divided into user and tester app, the server, and the blockchain.

The *upsi* architecture diagram is visualized in Figure 28.

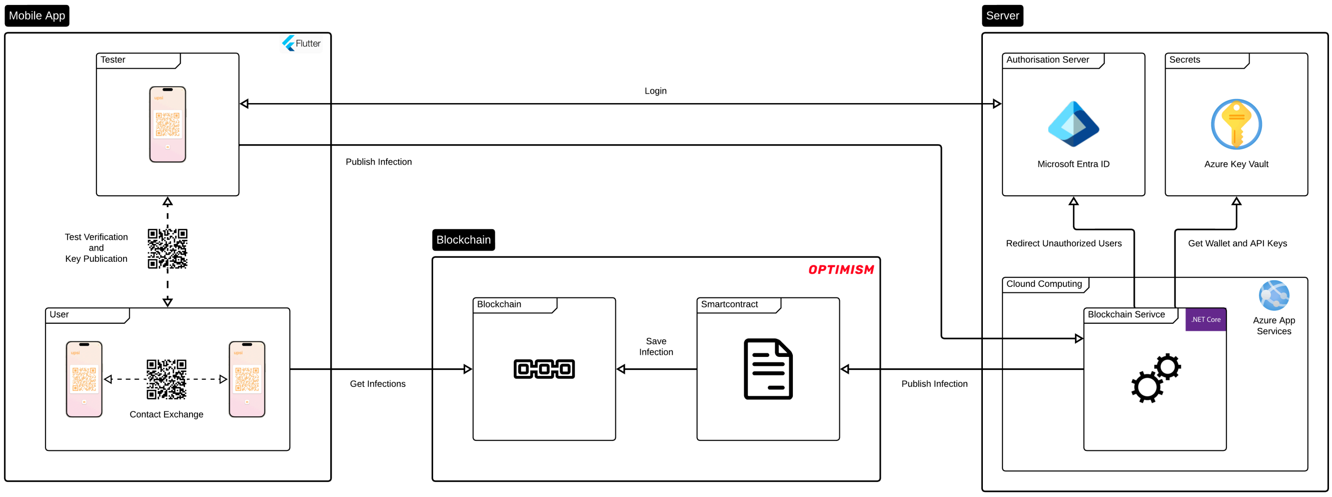


Figure 28: upsi *architecture diagram*

Source Code

The source code of all components is open source and can be viewd on GitHub:

- **Mobile App:** https://github.com/LaurinZubler/ba-app
- **Server:** https://github.com/LaurinZubler/ba-server
- **Blockchain**: https://github.com/LaurinZubler/ba-chain

5.3 Mobile App

Introduction

This section describes the mobile application technology evaluation and explains the two mobile applications in detail.

5.3.2 Mobile Technology Evaluation

5.3.2.1 Requirements

Cross-Platform

The mobile app should be used by as many people as possible, regardless of the smartphone they are using. The most used mobile operating systems are Android and iOS. The two are holding a global market share of around 99%[23]. Therefore, the mobile app should be compatible with these two operating systems.

In the mobile app development, several approaches can be taken to support both Android and iOS:

- **Single-Platform:** Two separate applications are developed in the native programming languages, one for Android and one for iOS. The advantages include performance, native look and feel, and straightforward access to device functions. The disadvantages are that the code needs to be written twice, maintenance is required for two codebases, and two technologies must be learned.
- **Hybrid:** Development is done using web technologies that are rendered on the device. The advantage is a single codebase and the technologies are already familiar to many developers. The disadvantage is poorer performance and restricted access to OS features.
- **Cross-Platform:** The code is written in a programming language that compiles to native code. The advantage is the look and feel of a native app. The disadvantage is somewhat restricted access to OS features.

For the mobile app, the **Cross-Platform** approch is chosen. The disadvantages regarding access to OS features are considered minor since the app only utilizes very standard OS features such as the camera or push notifications.

Feature Support

The mobile app must support following features either natively or through third-party libraries:

- Create and read QR codes
- Read data from blockchain
- Create and verify BLS digital signatures
- Receive push notifications
- Send HTTP requests
- Switch between multiple languages (l10n)

[23] https://de.statista.com/statistik/daten/studie/184335/umfrage/marktanteil-der-mobilen-betriebssysteme-weltweit-seit-2009/

Developer Friendly

The mobile app technology should meet some developer friendliness criteria:

- **Age:** The technology should neither be brand new, to avoid initial teething problems, nor too old, to prevent obsolescence and to benefit from recent technological advancements.
- **Support:** Documentation and tutorials should be available, either from the technology creators or through external sources. There should also be numerous questions on StackOverflow[24], promising quick help.
- **Popularity:** The technology should have gained some popularity. This can be observed by the activity on StackOverflow, as well as the number of stars, forks, and contributions on GitHub[25]. It should also perform well in the annual StackOverflow developer survey[26].

5.3.2.2 Evaluated Technologies

Information

In Table 2 the evaluated mobile app technologies are summarized. They are all Cross-Platform technologies and are supporting the required features.

	Launched By	Launch Date	Language
Flutter[27]	Google	2017	Dart
React Native[28]	Meta (Facebook)	2015	JavaScript JSX (HTML) CSS
Kotlin Multi Platform[29]	JetBrains	2017 Stable: Q4 2023	Kotlin
.NET Maui[30]	Microsoft	2022	C# XAML

Table 2: Evaluated mobile app technologies

[24] https://stackoverflow.com/
[25] https://github.com/
[26] https://survey.stackoverflow.co/2023/
[27] https://flutter.dev
[28] https://reactnative.dev/
[29] https://www.jetbrains.com/kotlin-multiplatform/
[30] https://learn.microsoft.com/en-us/dotnet/maui/

Metrics

The metrics from GitHub and StackOverflow used to assess the technology popularity and support are presented in Table 3. The data is collected at the beginning of March 2024.

	GitHub				StackOverflow	
	Stars	**Forks**	**Issues**[31]	**Pull Requests**[31]	**Questions**	**Survey**[32]
Flutter	161 k	26.3 k	10.5 k	12	Flutter: 175′086 Dart: 93′255	*Dart* Usage: 3.13% *Flutter* Usage: 9.21% Desired: 7% Admired: 59%
React Native	115 k	23.8 k	461	41	React Native: 136′593	Usage: 9.14%
Kotlin Multi Platform	47.2 k	5.5 k	N/A	142	Kotlin Multi Platform: 1535 Kotlin: 94′587	*Kotlin* Usage: 9.7% Desired: 12% Admired: 66.77%
.NET Maui	21.3 k	1.5 k	2.2 k	41	.NET MAUI: 6284 C#: 1′623′115	*.NET MAUI* Usage: 2.46% *C#* Usage: 29.16 % Desired: 21% Admired: 63%

Table 3: Evaluated mobile app technology GitHub and Stackoverflow metrics

[31] Open and older than 6 Months

[32] Admired vs Desired: https://survey.stackoverflow.co/2023/#technology-admired-and-desired

5.3.2.3 Desicion

Criteria

The decision is based on criteria that were derived from the requirements, with the inclusion of a personal preference for working with this technology.

The criteria are as follows:

- Age
- Support: Based on the StackOverflow metrics.
- Community: Based on the GitHub metrics.
- Programming Language Support: Based on the programming language StackOverflow metrics.
- Personal Preference

Rating

For each technology a value ranging from 1 to 3 is assigned to each criterion. These values are manually estimated based on the information presented in Table 2 and Table 3, without the use of complex calculations. Each criterion is assigned a multiplication factor to weight its importance.

Matrix

The decision matrix is displayed in Table 4, presenting the assigned values and multiplication factors, as well as the calculated results.

	Age	**Support**	**Community**	**Language Support**	**Personal Preference**	**Result**
Factor	1	0.5	0.5	0.5	1	
Flutter	3	3	3	2	3	10
React Native	3	2	3	3	1	8
Kotlin Multi Platform	1	1	1	2	2	5
.NET Maui	2	1	1	3	3	7.5

Table 4: Mobile app technology desicion matrix

Summary

The highest overall score is achieved by Flutter, resulting in its selection as the chosen mobile app technology.

5.3.3 Data Exchange Technology Evaluation

QR Code

QR codes are the technology chosen for data exchange between users and testers. QR codes are easy to implement for developers and have gained significant recognition through various applications, see chapter 2.3. Both sender and receiver remain anonymous using QR codes, and only the desired data is exchanged.

Evaluated Technologies

In addition to QR codes, the following technologies were evaluated:

- **NFC:** Near Field Communication[33] allows for the contactless exchange of data over short distances. NFC is used for contactless smartphone payments and therefore gained a certain level of popularity. The application in *upsi* would be very user friendly, as two smartphones simply could be held next to each other for the data exchange. NFC is not used in *upsi*, because the implementation into applications is very challenging, and two-way data exchange is not supported[34].
- **Bluetooth:** Data can be transmitted wirelessly via Bluetooth[35]. It is often used for pairing computers or smartphones with electronic devices and therefore gained a certain level of popularity. Bluetooth is not suitable for *upsi*, because pairing two smartphones requires some effort and the use of QR codes was considered more user-friendly.

5.3.4 User App

Key Handling

A new pair of private and public keys is periodically generated by the user app and stored locally on the device. The regeneration of keys is performed to enhance user privacy. When a public key of a user is made public, only the STI infections occurring during the validity period of the key can be attributed to the user. The validity period is set to 30 days, which is considered a reasonable timeframe.

Since the keys are used for BLS signatures, they consist of a 49-byte public key and a 32-byte private key. For easier development, the keys are stored as UTF-8 strings and not as hex byte arrays, requiring 98 bytes for the public and 64 bytes for the private key.

Example private key:

```
180d75e97a8d531ffd2e2daa0c5c47805439f62473f2bc17273361394804ba3c
```

Example public key:

```
0x94dbee4054676a2bd9fcc7c1d0d53bb4a62da076a1b91f34947b6115ba1a98b06d6a7c032393782
ce581cc1816274410
```

Contact Exchange

On the home screen of the user app, the contact exchange QR code is displayed, and the camera can be activated via a button to scan QR codes from other users. The contact exchange QR code contains the currently valid public key and a timestamp. The timestamp is verified by the recipient to ensure it is within a valid time period. Otherwise, the contact is

[33] https://en.wikipedia.org/wiki/Near-field_communication

[34] https://stackoverflow.com/questions/16712741/is-it-possible-to-make-two-way-communication-between-two-devices-via-nfc

[35] https://en.wikipedia.org/wiki/Bluetooth

discarded. This process ensures that no outdated and potentially invalid contacts are saved. The contact QR code is regenerated every 5 seconds and is valid for 15 seconds. If the contact is valid, it is stored locally on the device.

The contact exchange QR code format in JSON:

```json
{
    "type": "contact",
    "data": {
        "publicKey": "<public key>",
        "dateTime": "<UTC timestamp: yyyy-MM-ddTHH:mm:ss.SSSZ>"
    }
}
```

Screenshots of the home screen with the contact exchange QR code and the camera on are displayed in Figure 29 and Figure 30.

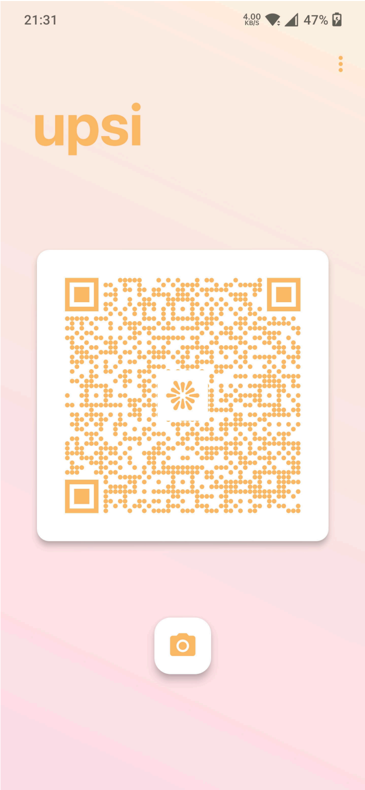


Figure 29: Screenshot of the home screen with contact exchange QR code

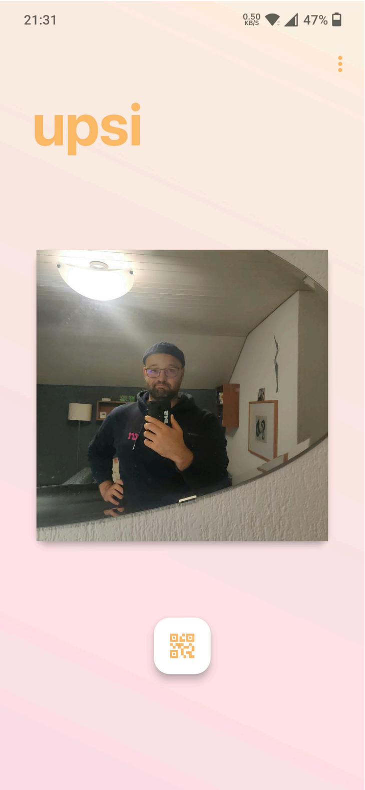


Figure 30: Screenshot of the home screen with camera activated

Infection Verification

In the event of a positive test result, an infection QR code can be generated by scanning the Proof of Attendance (PoA) QR code in the tester app. The infection QR code is then scanned by the tester to publish the infection.

Screenshot of the PoA QR code in the tester app and the infection QR code are displayed in Figure 31 and Figure 32.

The PoA QR code contains the test time and the SHA-256 hashed email of the tester. As with the contact exchange QR code, the PoA QR code is only valid for 15 seconds.

The PoA QR code format in JSON:

```
{
    "type": "poa",
    "data": {
        "tester": "<SHA-256 tester mail>" ,
        "testTime": "<UTC timestamp: yyyy-MM-ddTHH:mm:ss.SSSZ>"
    }
}
```

The infection QR code contains multiple public keys of the user, as well as the aggregated BLS signature. The infection is signed with all private keys corresponding to the public keys included in the infection event QR code. The BLS scheme allows for the publication of a single signature while still ensuring the validity of all keys. The message used for signing is the data contained in the PoA QR code, the hash of the tester, and the test timestamp.

The number of public keys determines how far back in time contacts can be to receive the partner notification. The number of public keys is always fixed. If the user does not have enough keys, new ones are generated. This approach enhances privacy by preventing inferences about app usage duration based on the number of keys. Due to the space limitations on the smartphone the infection QR is readable on the test device up to 7 public keys. This is furhter discussed in the limitations chapter 5.6.

Infection QR code format in JSON:

```
{
    "type": "infection",
    "data": {
        "infectee": [
            "<public key 1>",
            "<public key 2>",
            ...
            "<public key n>"
        ],
        "tester": "<SHA-256 tester mail>",
        "testTime": "<UTC timestamp: yyyy-MM-ddTHH:mm:ss.SSSZ>",
        "signature": "<BLS signature>"
    }
}
```

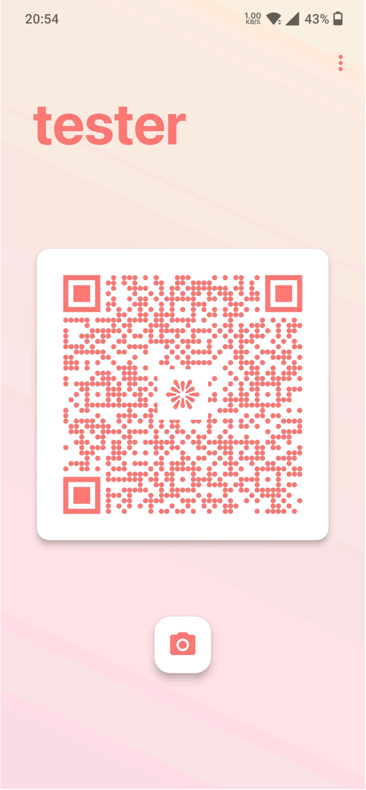


Figure 31: Screenshot of the tester app home screen with the PoA QR code

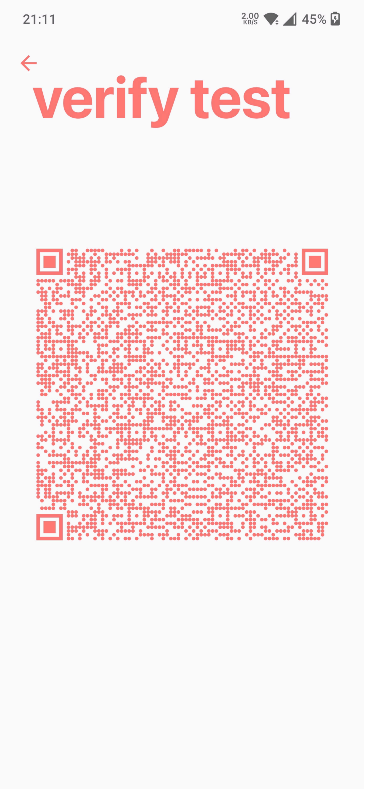


Figure 32: Screenshot of the infection QR code

Exposure Notification

Newly published infections are periodically read from the blockchain, to check for possible exposures. By verifying the signature, it is ensured that the infection is comming from a trusted tester and that the infected person attended the test.

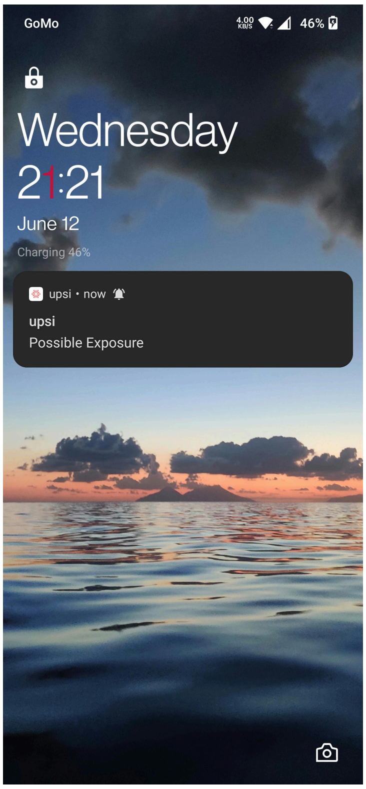
Each STI has a notification period. If the infected individual is in the saved contacts and within the notification period, the user is receiving a partner notification push on the smartphone. Upon the next app opening, an exposure warning is displayed on the home screen. Clicking on this warning is opening an info screen with details about the STI and guidance.

Screenshots of the push notification, the exposure warning and the STI info screen are displayed in Figure 33, Figure 34 and Figure 35.

Invented STIs are used during app development. For real STIs, the notification periods and info screen content should be defined by STI experts.

To start the periodic reading of the new infections on the blockchain, Flutter Workmanager was utilized[36]. The Workmanager allows for the execution of code in the background. With Android, a job can be executed every 15 minutes. Some limitations were observed with the Workmangager, further described in limitations chapter 5.6.

To access the blockchain, an API from Infura[37] is used. Infura is a managed service that provides access to various blockchain networks. With the free tier, it is possible to send 100,000 requests per day.



Wednesday
21:21
June 12
Charging 46%

upsi • now
upsi
Possible Exposure

Figure 33: Screenshot of the partner notification push

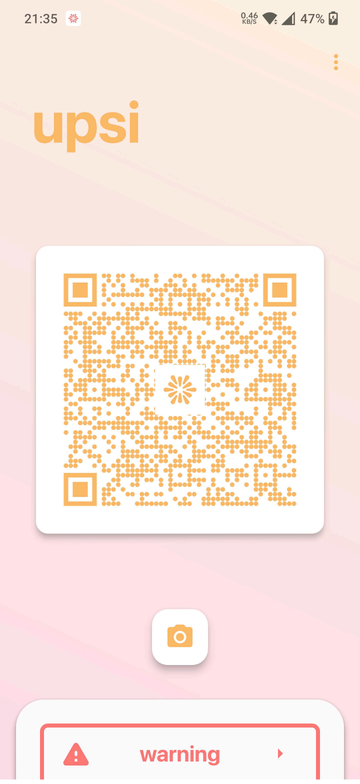
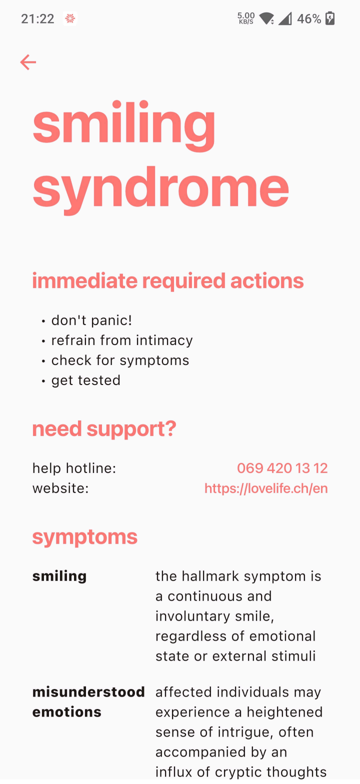


Figure 34: Screenshot of the home screen with the possible exposure warning



smiling syndrome

immediate required actions

- don't panic!
- refrain from intimacy
- check for symptoms
- get tested

need support?

help hotline:	069 420 13 12
website:	https://lovelife.ch/en

symptoms

smiling	the hallmark symptom is a continuous and involuntary smile, regardless of emotional state or external stimuli
misunderstood emotions	affected individuals may experience a heightened sense of intrigue, often accompanied by an influx of cryptic thoughts

Figure 35: Screenshot of the STI info screen

Multilingual The displayed texts in the app are available in multiple languages and are set based on the operating system's language. Texts are currently available in English and German. Only a small effort is required to add a new language by adding the translations in an `.arb`[38] file.

Secrets An API key is used to identify the app with Infura. The API key is secret and should not be published. To prevent it from being found in the source code, the API key is read from an environment `.env` file[39] that is not checked into the Git repository. However, the key could be extracted from the release binary. To prevent this code obfuscation[40] should be used when publishing the app.

[36] https://pub.dev/packages/workmanager
[37] https://www.infura.io/
[38] https://github.com/google/app-resource-bundle
[39] https://pub.dev/packages/flutter_dotenv
[40] https://docs.flutter.dev/deployment/obfuscate

5.3.5 Tester App

Test PoA

On the home screen of the tester app, the PoA QR code is displayed, and the camera can be activated via a button to scan the infection QR code from users.

This process is described in detail in chapter 5.3.4.

Publish Infection

The signed infection received from the user is sent to the server via an HTTP POST request.

Currently, the STI cannot be selected, and a random STI is chosen for the publication. This is furhter discussed in the limitations chapter 5.6.

Login

The tester authenticates with the server via login. Only registered testers can publish infections through the server.

The login feature has not been implemented. Further described in limitations chapter 5.6.

5.3.6 Design Prototype

Figma Screenshots

A design prototype for the user app was created with Figma[41] prior to implementation. Figma allows screens to be compiled into an interactive presentation. The *upsi* design prototype can be viewed here: https://www.figma.com/proto/RzrlqXuf1hCJBPQkBAe6Jn/upsi---design-prototype.

All created screens, including some color variations, of the design prototype are available in the appendix at chapter 11.5.

Color Scheme

The color scheme is based on two colors:

- Yellow

  🟧

  hex code: #FFB967
- Red

  🟥

  hex code: #FF7979

In the user app, the primary color is yellow. Warnings and exceptional actions, such as the exposure warning or the verify test screen, are displayed in red. In the tester app, the primary color is red.

[41] https://www.figma.com/

Material Design Material Design[42] components were used to achieve consistent and modern-looking screens.

5.3.7 Application Architecture

5.3.7.1 Clean Architecture

Clean Architecture The application architecture of the mobile apps is based on clean architecture[43] to ensure a clear separation of concerns, promote maintainability, and enhance testability.

The application is divided into four layers:

- **Domain Layer:** Containing the core business logic, including entities and services. It is isolated from other layers to ensure that the business rules are not affected by changes in the external environment.
- **Application Layer:** Handling the coordination of the domain logic.
- **Presentation Layer:** Containing screens and UI components.
- **Data Layer:** External data, such as APIs and local storage.

5.3.7.2 Core Package

Core Package Since the tester and user mobile app share some of the same code, these source files are extracted into a separate package, which is loaded as a dependency by both apps. This approach ensures a single codebase and eliminates duplicated code.

5.3.8 Libraries

Riverpod Framework The Riverpod framework[44] is used for easy state management and dependency injection across the application.

Freezed Freezed[45] is used to simplify the creation of classes and minimize boilerplate code by generating common features such as value equality, copy methods, and pattern matching.

5.3.9 Testing

Unit Tests The business logic is tested using unit tests to enhance code quality and reliability. Mocks are used to simulate dependencies, allowing each unit to be tested independently.

Unit test report of from the GitHub Pipeline can be found in the appendix at chapter 11.5.

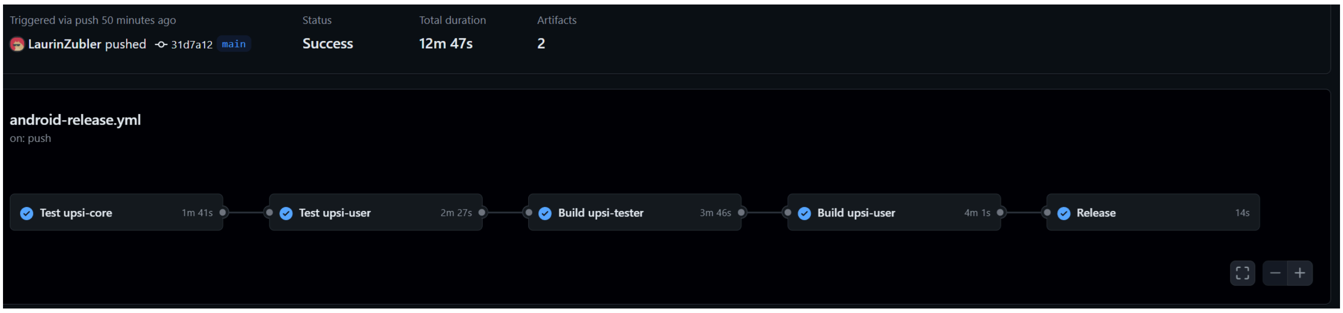
[42] https://m3.material.io/
[43] https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html
[44] https://riverpod.dev/
[45] https://pub.dev/packages/freezed

5.3.10 CI/CD

GitHub Pipline

A GitHub pipeline is set up to automatically test, build, and release the mobile apps upon push to the main branch.

A screenshot of a successfully run pipeline is diplayed in Figure 36.



Triggered via push 50 minutes ago
LaurinZubler pushed -o- 31d7a12 main

Status
Success

Total duration
12m 47s

Artifacts
2

android-release.yml
on: push

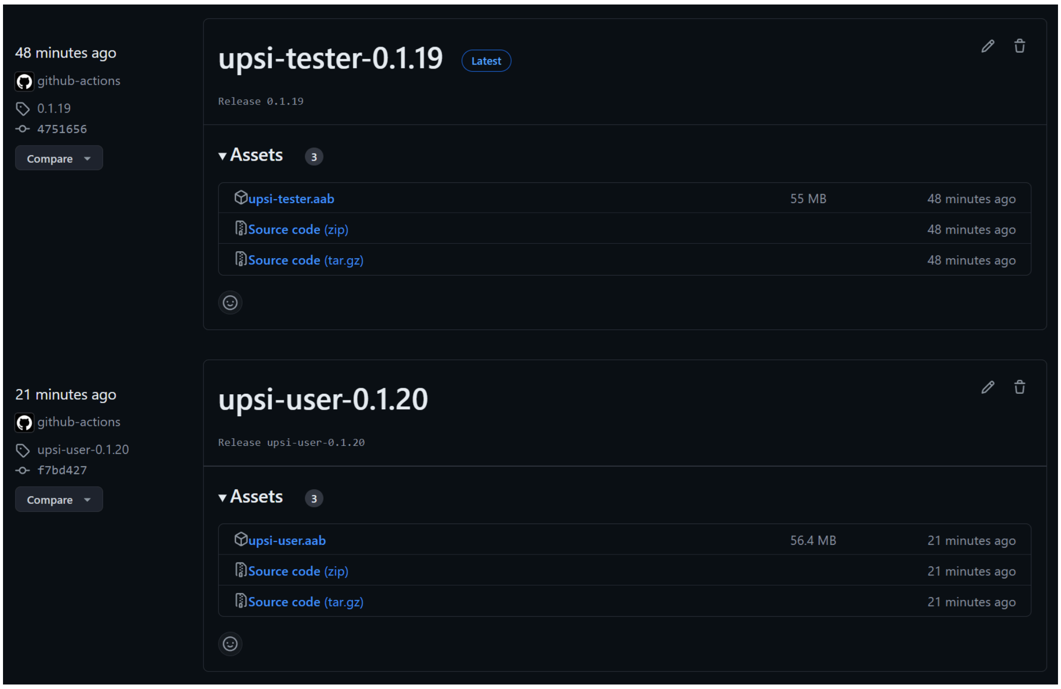
Test upsi-core 1m 41s
Test upsi-user 2m 27s
Build upsi-tester 3m 46s
Build upsi-user 4m 1s
Release 14s

Figure 36: Screenshot of a successfully run GitHub pipline for the mobile apps

Release

The GitHub pipline is creating a release with the android mobile apps on GitHub. The `.aab` files to install the android apps on a smartphone can be downloaded under: https://github.com/LaurinZubler/ba-app/releases

A sceenshot of an app release on GitHub is displayed in Figure 37.



48 minutes ago
github-actions
0.1.19
-o- 4751656
Compare

upsi-tester-0.1.19 Latest

Release 0.1.19

Assets 3

upsi-tester.aab	55 MB	48 minutes ago
Source code (zip)		48 minutes ago
Source code (tar.gz)		48 minutes ago

21 minutes ago
github-actions
upsi-user-0.1.20
-o- f7bd427
Compare

upsi-user-0.1.20

Release upsi-user-0.1.20

Assets 3

upsi-user.aab	56.4 MB	21 minutes ago
Source code (zip)		21 minutes ago
Source code (tar.gz)		21 minutes ago

Figure 37: Screenshot of GitHub releases with the android mobile apps

5.4 Blockchain

Introduction

This section describes the blockchain technology evaluation and explains the implemented smart contract in detail.

5.4.2 Technology Evaluation

5.4.2.1 Requirements

Consensus Mechanism

The consensus mechanism should be an energy efficient variant, such as Proof of Stake, rather than Proof of Work.

Smart Contracts

The blockchain should support smart contracts to facilitate the storage of infections on the blockchain.

Transaction Price

The blockchain gas fees should be low to minimize the cost of storing infection data.

Market Cap

The blockchain should have a high market cap to ensure its long-term viability.

Developer Friendly

The blockchain technology should meet several developer-friendly criteria:

- *Age:* The technology should not be brand new to avoid initial teething problems.
- *Support:* Documentation and tutorials should be available, either from the technology creators or through external sources.
- *Popularity:* The technology should have gained some popularity. This can be observed by the activity on StackOverflow, as well as the number of stars, forks, and contributions on GitHub[46].

[46] https://github.com/

5.4.2.2 Evaluated Technologies

Information In Table 5 the evaluated blockchain technologies are summarized. All of them are using an energy efficient consensus mechanism.

	Abbr.	Ranking[47]	Market Cap	Nodes	C. M.[48]	Contract Language	Gas Price	Speed
Ethereum[49]	ETH	1	$410B	PoS	Solidity	6902	high	to 5min
Solana[50]	SOL	5	$58.8B	PoS	Rust	N/A	low	instant
Polygon[51]	MATIC	15	$10B	PoS	Solidity	317	low	2.3s
Dfinity[52]	ICP	20	$6.1B	PoUW	Motoko	559	N/A	1s
Aptos[53]	APT	29	$4.2B	PoS	Move	400	low	5s
Optimism[54]	OP	31	$3.9B	Optimistic Rollup	Solidity	N/A	low	1s
Arbitrum[55]	ARB	46	$2.6B	Optimistic Rollup	Solidity	N/A	low	instant
Sui[56]	SUI	52	$1.9B	Delegated PoS	Move	451	low	instant
Tezos[57]	XTZ	77	$1.3B	PoS	LIGO	N/A	low	15s

Table 5: Evaluated blockchain technologies

[47] https://coinmarketcap.com/
[48] Consensus Mechanism
[49] https://ethereum.org
[50] https://solana.com/
[51] https://polygon.technology/
[52] https://dfinity.org/
[53] https://aptoslabs.com/
[54] https://www.optimism.io/
[55] https://arbitrum.io/
[56] https://sui.io/
[57] https://tezos.com/

GitHub Metrics The GitHub metrics used to assess the technology popularity and support is presented in Table 6. The data is collected at the beginning of March 2024.

	Stars	Forks	Issues[58]	Pull Requests[58]
Ethereum	45.3k	18.7k	196	33
Solana	11.5k	3.5k	470	0
Polygon	944	455	0	0
Dfinity	1.4k	279	N/A	N/A
Aptos	5.7k	3.4k	182	9
Optimism	4.9k	2.5k	30	1
Arbitrum	628	304	17	6
Sui	5.6k	10.8k	539	181
Tezos	309	N/A	2397	N/A

Table 6: Evaluated blockchain technology GitHub metrics

5.4.2.3 Desicion

Criteria The decision is based on criteria that were derived from the requirements, with the inclusion of a personal preference for working with this technology.

The criteria are as follows:

- Transaction Price
- Transaction Speed
- Marcet Cap
- Community: Based on the GitHub metrics.
- Personal Preference

Rating For each technology a value ranging from 1 to 3 is assigned to each criterion. These values are manually estimated based on the information presented in Table 5 and Table 6, without the use of complex calculations. Each criterion is assigned a multiplication factor to weight its importance.

[58] Open and older than 6 Months

Matrix

The decision matrix is displayed in Table 7, presenting the assigned values and multiplication factors, as well as the calculated results.

	Price	Speed	Marcet Cap	Community	Personal Preference	Result
Factor	1.5	0.5	1	1	1.5	
Ethereum	1	1	3	3	3	12.5
Solana	3	3	3	2	2	14
Polygon	3	2	3	1	3	14
Dfinity	3	3	3	1	1	11.5
Aptos	3	2	2	2	2	12.5
Optimism	3	3	2	2	3	14.5
Arbitrum	3	3	2	1	3	13.5
Sui	3	3	1	2	2	12
Tezos	3	1	1	1	1	8.5

Table 7: Blockchain technology desicion matrix

Summary

The highest overall score is achieved by Optimism, resulting in its selection as the chosen blockchain technology. Optimism is an Ethereum Layer 2 blockchain.

5.4.3 Smart Contract

Deployment

The smart contract is deployed to the Optimism Sepolia testnet and all transactions can be viewed on: https://sepolia-optimistic.etherscan.io/address/0x5059d4FC4e72C7f5dA98be7e32BA1F9a16546904

DAO

As only the contract owner can grant the event emitter role, and therefore decides which test centers can publish infections, a concentration of power is held by the contract owner. This contradicts the fundamental principle of decentralization. However, it is essential to ensure that only verified test centers can publish infections.

This power of the owner can be weakened by transferring the contract ownership to a decentralized autonomous organization (DAO). Within the DAO, multiple actors could democratically decide on the inclusion of new test centers. These DAO actors could be the already verified test centers.

5.4.3.2 Functionalities

Emit Infection Event

The main function of the smart contract is to emit an infection event.

The function's parameters are:

- Infection
- Infectees
- Tester
- Test time
- Signature

The test time is saved as a Unix timestamp using a 256-bit unsigned integer, representing the number of seconds passed since 01.01.1970. Strings are used for the other parameters to simplify development. As all parameters contain hex values, byte arrays could be used to reduce storage size and lower the gas fee.

The function emits an `InfectionEvent` event, making the infection information publicly accessible on the blockchain.

Event Emitter Role

The Event Emitter Role is defined by the contract and only addresses with this role can emit infection events. This role can be granted or revoked for specific addresses by the contract owner.

This is ensuring that only trusted test centers can publish infections.

Ownership

The ownership of the contract can be transferred by the current owner to a new address, making the new address the contract owner.

5.4.3.3 Development

Hardhat

Hardhat[59] was used as a development environment for the smart contract. It facilitated unit testing, deployment, and contract verification.

OpenZeppelin

OpenZeppelin[60] is providing libraries for smart contracts. The Ownable and AccessControl libraries were utilized.

[59] https://hardhat.org/

[60] https://www.openzeppelin.com/

5.5 Server

Server

This section describes the blockchain technology evaluation and explains the server application in detail.

5.5.1 Technology Evaluation

.NET Core C#

An extensive technology evaluation was not conducted for the server. .NET Core with C# was selected directly due to the author's preference over Java and Node.js. The author had never worked with .NET C# before and wanted to gain experience with this technology.

5.5.2 Functionality

Publish Infection Event

The server is providing an API for the testers to write infection to the blockchain. The API can only be called by registered and logged-in testers. Wallet addresses for publishing the infection to the blockchain are stored in a Azure Secure Vault. Since the tester is identified through the login, the associated test center is known, and the corresponding wallet address is selected. The login feature has not been implemented. This is furhter discussed in the limitations chapter 5.6.

5.5.3 CI/CD

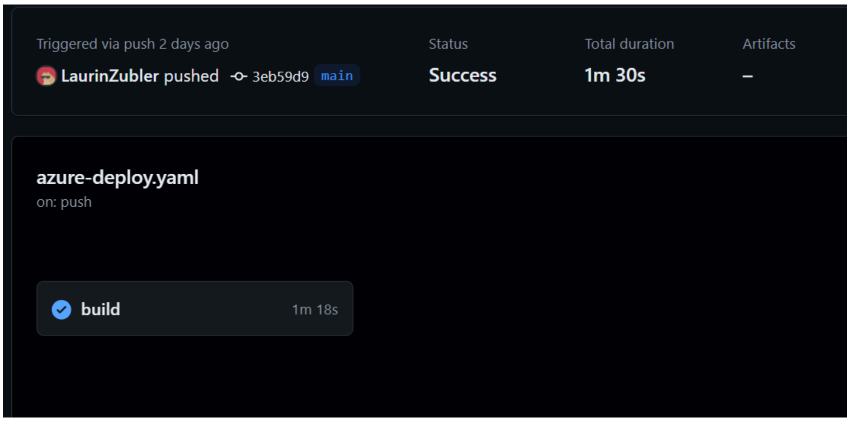
Azure

The server application is deployed to Azure App Services. Azure was selected as the cloud provider because the university is using Azure and covers the hosting costs.

GitHub pipeline

A GitHub pipeline is set up to automatically build and deploy the server application to Azure upon push to the main branch.

A screenshot of a successfully run pipeline is diplayed in Figure 38.



Triggered via push 2 days ago	Status	Total duration	Artifacts
LaurinZubler pushed 3eb59d9 main	Success	1m 30s	–

azure-deploy.yaml
on: push

build 1m 18s

Figure 38: Screenshot of a successfully run GitHub pipline of the server application

5.6 Limitations

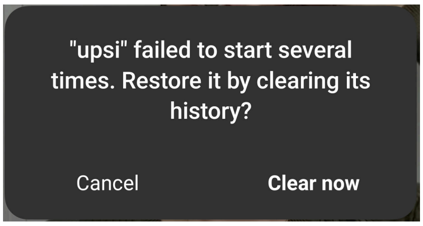
Introduction

Some features could not be implemented due to time constraints. The user mobile app and core functionalities were prioritized to ensure the application's primary objectives are met.

Trigger Blockchain Reading

To start the periodic reading of new infections on the blockchain, Flutter Workmanager[61] was utilized. This solution should be replaced by another method, as the performance of the Workmanager on iOS devices is uncertain[62], and on Android devices, a job can only be executed every 15 minutes. For demonstration and testing purposes, it would be beneficial to read the blockchain more frequently.

An attempt was made to register multiple background jobs with an initial delay so that the reading is triggered every 15 seconds. However, this caused the app to crash. A screenshot of the error message is presented in Figure 39.



"upsi" failed to start several times. Restore it by clearing its history?

Figure 39: Screenshot of the popup indicating the app crash

The originally intended solution to trigger the blockchain reading involved a server service that periodically sends a silent push notification. Upon receiving the silent push, the new infections are read. This was not implemented due to time constraints.

The system architecture with a silent push is illustrated in Figure 40.

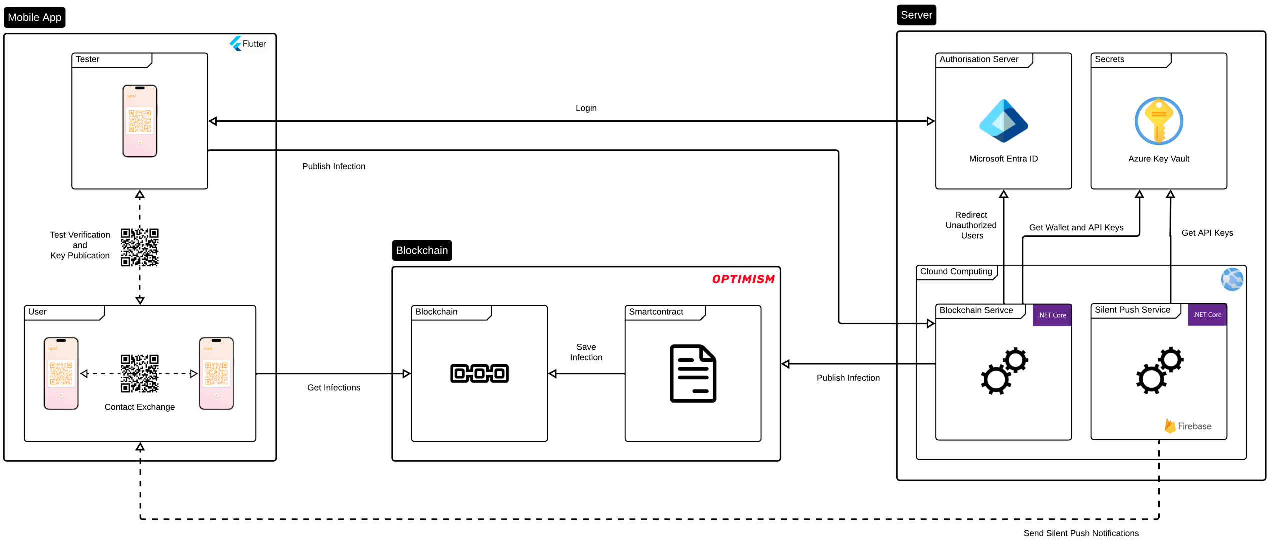


Figure 40: System architecture using silent push notifications

[61] https://pub.dev/packages/workmanager

[62] https://medium.com/vrt-digital-studio/flutter-workmanager-81e0cfbd6f6e

Infection QR Code

The infection QR code contains multiple public keys of the infected person to cover a certain time period in which contacts get notified. Since the size of the QR code is limited by the size of the smartphone screen, the QR code becomes smaller as more public keys are included. Beyond a certain number, the QR code cannot be read by the tester's smartphone.

With the developer smartphone, up to 7 keys in the QR code could be read. However, this can vary on other smartphones depending on the screen size and camera. To ensure reliability, the number of public keys has been set to 5. With the current setting with new keys every 30 days and 5 keys per infection, the notification period is approximately 5 months. Whether this is sufficient must be assessed by STI experts.

If the notification period needs to be extended, it is necessary to find a different technical solution. Here are some suggestions:

- **Multiple QR Codes:** The infection data can be split and transmitted to the tester in several steps. This process would have to be reflected in the apps accordingly, in a user-friendly way.
- **Colored QR Codes:** [24] The storage capacity of a QR code can be increased by using multiple colors instead of just black and white. This makes the QR code smaller and easier to read.
- **Reduce Data:** Instead of encoding the QR code data in UTF-8 JSON, it could be represented with a more data-efficient encoding, such as Binary JSON[63]. This makes the QR code smaller and easier to read.

Exposure Warning

In the screen design prototype, it was planned to display the exposure warning in an expandable bottom container. When expanded, all infection exposure warnings are shown. This component introduces some complexity as a dynamic list is displayed in a resizable container.

During implementation, this could not be achieved due to a limited understanding of Flutter UI development. The feature was prioritized as low-level and replaced with a simple button. Upon clicking, the info screen for the most recently received STI is appearing.

With more time for debugging, the implementation of the expandable bottom container is certainly possible.

Login

The login functionality was not implemented. Currently, the API for publishing infections is public. Since only the testnet of the blockchain is being used, this is considered not particularly problematic.

Published Infection

At the moment, an STI is selected randomly before publishing the infection, and the tester cannot select the STI. The process should be revised with the involvement of testing personnel to achieve a user-friendly solution.

[63] https://en.wikipedia.org/wiki/BSON

iOS

During development, the focus was on Android devices. iOS apps can only be built with a Mac computer, and neither an iOS smartphone nor a Mac computer was available. Since Flutter was used, extending the app to iOS should require minimal effort.

6 Conclusion

Preamble This chapter discusses the achieved results during this bachelor's thesis and suggests steps to develop *upsi* further.

6.2 Results

Introduction In this section, the results of the bachelor's thesis are further detailed.

Objectives This bachelor's thesis successfully meets the primary objectives described in chapter 1.3.

- A research was conducted to understand the technical approaches used in COVID-19 tracing apps and to identify existing technical solutions for STI partner notification. The findings were incorporated into the solution design.
- An STI tracing solution was implementend based on the the DP-3T protocol.
- An evaluation was conducted by consulting experts in the field of STIs to assess the feasibility and significance of the app.

Solution The implemented solution successfully meets the system requirements described in chapter 4.3.2.

- Privacy, decentralization, and anti-authoritarian principles are achieved by using the DP-3T protocol, further improved by utilizing blockchain technology for infection publication.
- User-friendliness is ensured by letting the test center handle the blockchain interaction and paying for the gas fees.
- Trustworthy partner notifications are ensured by the proof of test attendance signed by the user and by only permitting trusted test centers to publish infections.
- Openness is achieved by making all source code publicly available on GitHub.
- Consent is ensured by using QR codes for data exchange between devices, requiring explicit user action.

Evaluation

Valuable feedback was provided by the STI experts, which was incorporated into the development of *upsi*.

The key points are further elaborated here:

- **General positive:** Great interest and positivity were shown by the experts towards the idea of an STI tracing app. However, concerns were raised regarding the societal and integration hurdles of a potential implementation.
- **Anonymous STI rapid test:** It was not initially known that certain STI tests are conducted as rapid tests at test centers and that no data about the tested individuals is collected. This led to a revision of the concept and the introduction of the tester mobile app. Previously, a web application or integration into the test center IT system was considered. The final solution with the tester app is now regarded as much more refined and better adapted to real world conditions.
- **Stigmatisation:** STIs and partner notifications are still stigmatized in society. The experts argued that an STI tracing app would only be effective in an informed and sex-positive community. Men who have sex with men were proposed as such a community. Whether the app would be widely used by the general population remains questionable.
- **Event QR code:** It was pointed out by the experts that in certain situations, smartphones are not desired or do not function properly due to the conditions. Dark rooms and saunas used by men who have sex with men were mentioned. As a solution to this concern, the idea of an event QR code was proposed. Similar to the COVID-19 pandemic, QR codes would be provided for an event and scaned by the attendees. In the event of an STI infection, the event QR code would be published along with the public keys.
- **Test center landscape:** The previously made assumption that there are many different test centers, each using their own IT systems, was confirmed by the experts. These IT systems are often very old and legacy. Due to federal data protection regulations, the development and further development of such systems is cumbersome.

6.3 Further Work

Introduction

In this section, the further suggested work is detailed.

6.3.2 App

Features

To enhance user-friendliness and functionality, it is proposed to expand the app with several features.

The suggested features are:

- **Smartphone change:** Since the contacts and keys are stored on the device, it is currently not possible to switch smartphones. This can be addressed with an export and import function for contacts and keys. To reduce key data, a master key should be used from which all further private and public keys are derived.

  It must be considered how this can be implemented securely and cannot be used for misuse. This feature is working if both the old and new smartphones are available. If the smartphone is lost, all keys and contacts are also lost due to decentralized data storage on the device.
- **Login:** The user mobile app should be enhanced with an optional login, for example, using biometric data. This would further increase security and privacy.
- **Guidance:** To improve user-friendliness, informational texts and instructions should be integrated.
- **STI Infos:** The STI specific information and functionalities should be reviewed and updated with the help of STI experts.
- **Limitations:** The functions that were not implemented due to time constraints should be developed. More information on this can be found in chapter 5.6.

iOS

To increase the app’s reach, it should be extended to iOS devices. Since Flutter was used, this should require minimal effort.

6.3.3 Integration

Laboratory Tests

The introduction of *upsi* only makes sense if laboratory tests can also be covered. The system should be extended in collaboration with test centers, as integration into their IT systems is necessary to store user information between testing and receiving laboratory results.

Admin Portal

A test center admin portal should be developed to manage tester accounts and include a function to top up wallet balances with cryptocurrency to pay for the gas fees.

6.3.4 Study

Acceptance A comprehensive study should be conducted in collaboration with STI researchers to evaluate the effectiveness and acceptance of *upsi*.

7 Glossary

STI	Sexually Transmitted Infection. Further described in chapter 1.2.2
COVID-19	Coronavirus Disease 2019. https://en.wikipedia.org/wiki/COVID-19
SARS-CoV-2	Severe Acute Respiratory Syndrome Coronavirus 2. https://en.wikipedia.org/wiki/SARS-CoV-2
UN	United Nations. https://www.un.org/
WHO	World Health Organization. An United Nations agency. https://www.who.int/
HIV	Human Immunodeficiency Viruses. https://en.wikipedia.org/wiki/HIV
AIDS	Acquired immunodeficiency syndrome. https://en.wikipedia.org/wiki/HIV/AIDS
FOPH	Swiss Federal Office of Public Health. https://www.bag.admin.ch/
BAG	Bundesamt für Gesundheit Schweiz. https://www.bag.admin.ch/
Bluetooth	A short-range wireless technology. https://en.wikipedia.org/wiki/Bluetooth
GPS	Global Positioning System. https://en.wikipedia.org/wiki/Global_Positioning_System
DAO	Decentralized autonomous organization. Further described in chapter 2.2
QR Code	Quick Response Code. Furhter described in chapter 2.3
BLS	BLS digital signature. Futher described in chapter 2.4
SBB	Schweizerische Bundesbahnen (en: Swiss federal railways) https://www.sbb.ch/
Twint	Swiss cashless payment system. https://www.twint.ch/
PoW	Proof of Work. Furhter described in …
PoS	Proof of Stake. Furhter described in …
Blockchain	Decentralized distributed database. Further described in chapter 2.2
Hash	Hash function: https://en.wikipedia.org/wiki/Hash_function
Smart Contract	Executable code on a blockchain. Furhter described in chapter 2.2
Solidity	Programming language for smart contracts. Furhter described in chapter 2.2
Gas	… Furhter described in chapter 2.2
Fees	… Furhter described in chapter 2.2
L2	Layer 2 blockchain. Furhter described in chapter 2.2
MSM	Men who have sex with men
YMSM	Young men who have sex with men

PrEP	Pre-exposure prophylaxis for HIV prevention. https://en.wikipedia.org/wiki/Pre-exposure_prophylaxis_for_HIV_prevention
AI	Artificial Intelligence. https://en.wikipedia.org/wiki/Artificial_intelligence
Geosocial Networking App	Location based social networks. https://en.wikipedia.org/wiki/Geosocial_networking
TEK	Temporary Exposure Key. Furhter described in chapter 3.5.2.1.3
RPI	Rolling Proximity Identifier. Furhter described in chapter 3.5.2.1.3
JSON	JavaScript Object Notation. https://en.wikipedia.org/wiki/JSON
SHA-265	Secure Hash Algorithm. https://en.wikipedia.org/wiki/SHA-2
UTC	Coordinated Universal Time. https://en.wikipedia.org/wiki/Coordinated_Universal_Time

8 Bibliography

[1] World Health Organization, “Sexually transmitted infections (STIs)”. Accessed: May 20, 2024. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/sexually-transmitted-infections-(stis)

[2] World Health Organization, “Global progress report on HIV, viral hepatitis and sexually transmitted infections, 2021. Accountability for the global health sector strategies 2016-2021: actions for impact”, World Health Organization, Geneva, Switzerland, 2021. [Online]. Available: https://www.who.int/publications/i/item/9789240027077

[3] Bundesamt für Gesundheit, “Sexuell übertragene Infektionen und Hepatitis B/C in der Schweiz im Jahr 2022: eine epidemiologische Beurteilung”, Nov. 2023. [Online]. Available: https://www.bag.admin.ch/bag/de/home/das-bag/publikationen/periodika/bag-bulletin.html

[4] A. Ferreira, T. Young, C. Mathews, M. Zunza, and N. Low, “Strategies for partner notification for sexually transmitted infections, including HIV”, *Cochrane Database of Systematic Reviews*, no. 10, Oct. 2013, doi: 10.1002/14651858.CD002843.pub2.

[5] P. Daniore, T. Ballouz, D. Menges, and V. von Wyl, “The SwissCovid Digital Proximity Tracing App after one year: Were expectations fulfilled?”, *Swiss Medical Weekly*, vol. 151, no. 3536, p. w30031, Sep. 2021, doi: 10.4414/SMW.2021.w30031.

[6] S. Vaudenay, “Centralized or Decentralized? The Contact Tracing Dilemma”, in Cryptology ePrint Archive. 2020/531. 2020. [Online]. Available: http://infoscience.epfl.ch/record/277809

[7] A. Mendi and A. Cabuk, “Evaluation of Advantages and Creative Aspects of Blockchain Architecture”, in *1st International Symposium On Information Science And Technologies*, 2018.

[8] J. G. Keogh, A. Rejeb, N. Khan, and K. Zaid-Kaylani, “Chapter 68 - Blockchain: an enabler for safe food in global supply networks”, *Present Knowledge in Food Safety*. Academic Press, pp. 1045–1066, 2023. doi: https://doi.org/10.1016/B978-0-12-819470-6.00008-1.

[9] A. A. Monrat, O. Schelén, and K. Andersson, “A Survey of Blockchain From the Perspectives of Applications, Challenges, and Opportunities”, *IEEE Access*, vol. 7, pp. 117134–117151, 2019, doi: 10.1109/ACCESS.2019.2936094.

[10] Ethereum, “What is layer 2?”. Accessed: May 23, 2024. [Online]. Available: https://ethereum.org/en/layer-2/

[11] S. Hassan and P. D. Filippi, “Decentralized Autonomous Organization”, *Internet Policy Review*, vol. 10, no. 2, 2021, doi: 10.14763/2021.2.1556.

[12] Ethereum, “Decentralized autonomous organizations (DAOs)”. Accessed: May 23, 2024. [Online]. Available: https://ethereum.org/en/dao/

[13] M. Alharby and A. van Moorsel, “Blockchain-based Smart Contracts: A Systematic Mapping Study”, *Fourth International Conference on Computer Science and Information Technology (CSIT-2017)*, Aug. 2017, [Online]. Available: https://doi.org/10.48550/arXiv.1710.06372

[14] D. Boneh, B. Lynn, and H. Shacham, “Short Signatures from the Weil Pairing”, *Journal of Cryptology*, vol. 17, no. 4, pp. 297–319, 2004, doi: 10.1007/s00145-004-0314-9.

[15] X. Yan *et al.*, “Protecting Men Who Have Sex With Men From HIV Infection With an mHealth App for Partner Notification: Observational Study”, *JMIR mHealth and uHealth*, vol. 8, p. e14457, 2020, doi: 10.2196/14457.

[16] A. Liu *et al.*, “Developing a mobile app to support linkage to HIV/STI testing and PrEP for young men who have sex with men: LYNX pilot study protocol (ATN 140) (Preprint)”, *JMIR Research Protocols*, vol. 8, 2018, doi: 10.2196/10659.

[17] A. Liu *et al.*, “Developing a mobile app to support linkage to HIV/STI testing and PrEP for young men who have sex with men: LYNX pilot study protocol (ATN 140)”. [Online]. Available: https://clinicaltrials.gov/study/NCT03177512

[18] A. Y. Liu *et al.*, “DOT Diary: Developing a Novel Mobile App Using Artificial Intelligence and an Electronic Sexual Diary to Measure and Support PrEP Adherence Among Young Men Who Have Sex with Men”, *AIDS and Behavior*, vol. 25, no. 4, pp. 1001–1012, Apr. 2021, doi: 10.1007/s10461-020-03054-2.

[19] D. Levine, A. J. Woodruff, A. R. Mocello, J. Lebrija, and J. D. Klausner, “inSPOT: the first online STD partner notification system using electronic postcards”, *PLoS medicine*, vol. 5, no. 10, p. e213, Oct. 2008, doi: 10.1371/journal.pmed.0050213.

[20] J. Pellowski, C. Mathews, M. O. Kalichman, S. Dewing, M. N. Lurie, and S. C. Kalichman, “Advancing Partner Notification Through Electronic Communication Technology: A Review of Acceptability and Utilization Research”, *Journal of Health Communication*, vol. 21, no. 6, pp. 629–637, 2016, doi: 10.1080/10810730.2015.1128020.

[21] M. G. Contesse *et al.*, “Acceptability of Using Geosocial Networking Applications for HIV/Sexually Transmitted Disease Partner Notification and Sexual Health Services”, *Sexually Transmitted Diseases*, vol. 47, no. 1, Jan. 2020, doi: 10.1097/olq.0000000000001089.

[22] Fraunhofer AISEC, “Pandemic Contact Tracing Apps: DP-3T, PEPP-PT NTK, and ROBERT from a Privacy Perspective”. [Online]. Available: https://eprint.iacr.org/2020/489

[23] SwissCovid, “SwissCovid Components”. Accessed: Jun. 02, 2024. [Online]. Available: https://github.com/SwissCovid/swisscovid-doc/blob/main/01-architecture-overview.md

[24] M. E. Vizcarra Melgar, A. Zaghetto, B. Macchiavello, and A. C. A. Nascimento, “CQR codes: Colored quick-response codes”, in *2012 IEEE Second International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, 2012, pp. 321–325. doi: 10.1109/ICCE-Berlin.2012.6336526.

9 List of Figures

[64] https://www.lsvd.de/de/ct/1245-LGBT-Rechte-weltweit
[65] https://money.com/what-is-blockchain/
[66] https://www.rts.ch/info/suisse/11154401-les-suisses-se-sont-rues-sur-les-billets-degriffes-en-2019.html
[67] https://www.twint.ch/geschaeftskunden/unsere-loesungen/qr-code-sticker/
[68] https://www.bit.admin.ch/bit/de/home/themen/stories/covid-zertifikat.html
[69] https://inevitableeth.com/home/concepts/bls-signatures
[70] https://www.cpzh.ch/angebote/online-resultate/
[71] https://www.cpzh.ch/angebote/online-resultate/
[72] https://sxt.health/uk/pn/about
[73] https://www.thedramadownunder.info/let-them-know/
[74] https://www.thedramadownunder.info/let-them-know/
[75] https://www.thedramadownunder.info/let-them-know/
[76] https://lovelife.ch/en/protection/communicating-in-sexual-relationships
[77] https://x.com/BAG_OFSP_UFSP/status/1265979020585435137

10 List of Tables

11 Appendix

11.1 Disclaimer

Generative AI Generative AI was used during this bachelor’s thesis for coding and writing assistance.

Translation Tools Translation tools were used during the writing of this document to overcome language barriers.

Grammar Checkers Grammar checkers were used during the writing of this document to make it more pleasant to read. However, typos cannot be ruled out and are all original works by the author.

11.2 Project Planning

Introduction

In this section, the organisation and planning of the thesis is further detailed. Covered are the methodology, the tools utilized for planning and execution, and the delineation of project phases and milestones. Additionally, the implementation of Scrum and time tracking are discussed.

11.2.2 Method

Scrum+

In this project Scrum+ is employed, a hybrid approach blending elements of Scrum and the Rational Unified Process (RUP). From RUP, the concept of dividing the project into distinct phases is adopted, providing a structured framework for the entire project timeline. For more immediate and flexible planning, the agile methodologies of Scrum are integrated, particularly through the use of iterative sprints. This combination allows for both long-term structural clarity and short-term adaptability in the project management.

11.2.3 Tools

Jira

As the primary tool for the project management *Jira*[78] by *Atlassian* is used. It is specifically configured to align with the project's requirements, facilitating effective planning and tracking throughout the project's duration.

Jira Extensions

For a better experience and additional functions in *Jira* the following extensions are integrated:

- **Time Tracking:** For more accurate time tracking, *Timesheet Tracking for Jira* by *TouchDown* is used.[79]
- **Risk Management:** For risk tracking and analysis *Hedge: Risk Management, Risk Register & Risk Matrix for Jira* from *Appfire* is employed.[80]

[78] https://www.atlassian.com/software/jira

[79] https://marketplace.atlassian.com/apps/1216988/timesheet-tracking-for-jira

[80] https://marketplace.atlassian.com/apps/1227408/hedge-risk-management-risk-register-risk-matrix-for-jira

11.2.4 Phases

Epics

The project is initial segmented into rough phases, which are implemented using *Jira* Epics. Five phases are defined to structure the project's timeline and key stages.

The epics at the project start are listed in Table 8. The to phases *Development* and *Evaluation* are overlapping in time.

Phase	Duration	Description
Project Setup	2 Weeks	Setup project management tools and documentation
Analysis	2 Weeks	Conduct literature research and user study
Development	7 Weeks	Develop a prototype
Evaluation	7 Weeks	Collect expert feedback and evaluate prototype
Project Completion	2 Weeks	Complete documentation and submission

Table 8: Planned project phases at project start

For ongoing task during the whole project the two epics *Documentation* and *Project Management* are created.

A screenshot of the *Jira* timeline captured at the start of the project is presented in Figure 41, illustrating how these phases were organized and visualized within the tool.

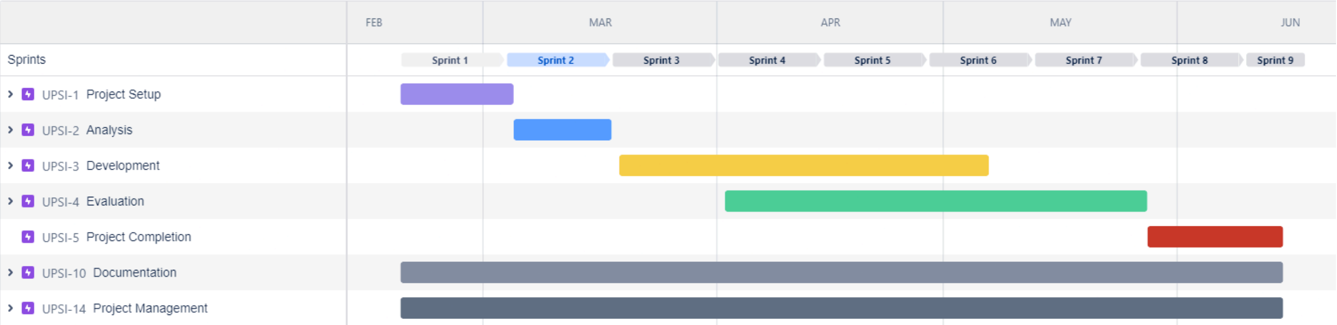


Figure 41: Screenshot Jira *timeline. Planned project phases at the beginning of the project*

11.2.5 Milestones

Introduction

Milestones are significant markers that denote critical achievements and points of progress in a project's timeline. For this project, a milestone was defined at the end of each phase. Since *Jira* is not offering a native functionality for creating milestones, a new issue type is created for this purpose. Similar to tasks, these milestones are initially placed in the backlog and then added to sprints as the project is progressing. They are not only serving as goals to strive for but also as opportunities to evaluate the project's direction and success at various stages.

11.2.5.2 End of Project Setup

Due Date 04.03.2024

Actual achieved 04.03.2024

Acceptance Criterias

- Project management tool set up
- Project time tracking tool set up
- Project documentation document set up
- Automated CI to publish project documentation
- Automated CI to publish meeting minutes
- Project plan: How, when and who will work on the project?
- Risks: What can endager the project and how to handle it?
- Quality assurance: How is a good quality of the project ensured?

11.2.5.3 End of Analysis

Due Date 18.03.2024

Actual achieved 19.05.2024

Acceptance Criterias

- Literature Review: What are STI's and how does tracing work?
- Technology Evaluation: What mobile platform and blockchain should be used for the implementation?
- Evaluation: Which test center organisations can be contacted?

11.2.5.4 End of Development

Due Date 05.05.2024

Actual achieved 14.06.2024

Acceptance Criterias

- Prototype implemented

11.2.5.5 End of Evaluation

Due Date	27.05.2024
Actual achieved	17.05.2024
Acceptance Criterias	• Expert feedback on prototype collected.

11.2.5.6 Project Completion

Due Date	14.06.2024
Actual achieved	14.06.2024
Acceptance Criterias	• Documentation complete • All relevant elements submitted on time

11.2.6 Scrum Elements

Introduction In this section, the applied scrum elements are furhter detailed.

Sprint The tasks are completed in two-week sprints.

During the ongoing sprint the tasks were managed using the *jira* sprint board. In four swimlane the progress of the tasks are visualized.

These swimlanes are used:
- **To Do** For tasks that have not yet been started
- **In Progress** For tasks that are currently being worked on
- **Review** For tasks waiting for a review from the advisor
- **Done** For tasks that are done

Backlog Refinement In the backlog refinement new tasks are created in the backlog and the time for completion is estimated.

Sprint Planning Before each sprint start, the sprint is filled with tasks from the backlog so that the estimated time of work is 2 weeks.

Sprint Review At the end of each sprint, the outcome of the Sprint is inspected and future adaptations are determinated. The Sprint review is part of the weekly meeting with the advisor.

Daily Scrum Meeting Since the project team only consists of one person, no daily Scrum meetings are held.

Weekly Meeting with Adviser Every week a meeting with the project advisor is held.

11.2.7 Time Tracking

Time Expenditure

The bachelor thesis is worth 12 ects credits, each credit is valued with 30 hours of work. This results in a total workload of 360 hours. Distributed over the 17 weeks of the semester a working time of approximately 21 hours or 2.6 days is targeted.

Tracking

To ensure sufficient effort is put into this thesis, working hours are recorded throughout the semester.

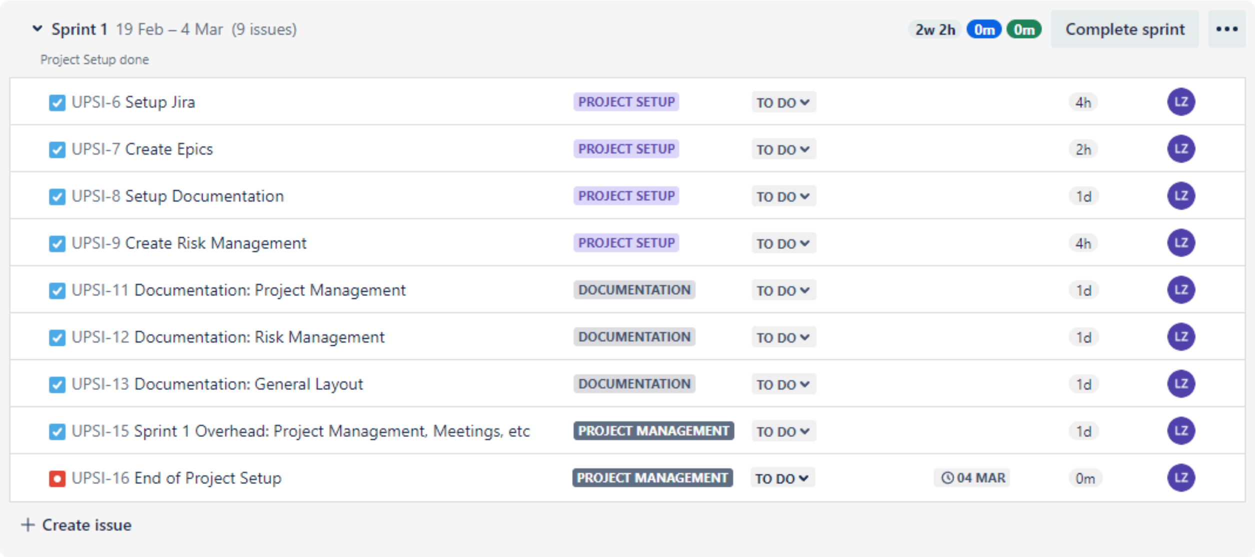
A total of 425 hours were spent at the end of the thesis.

11.2.8 Sprints

11.2.8.1 Sprint 1

Period 19.02.2024 - 04.03.2023

Screenshot Jira

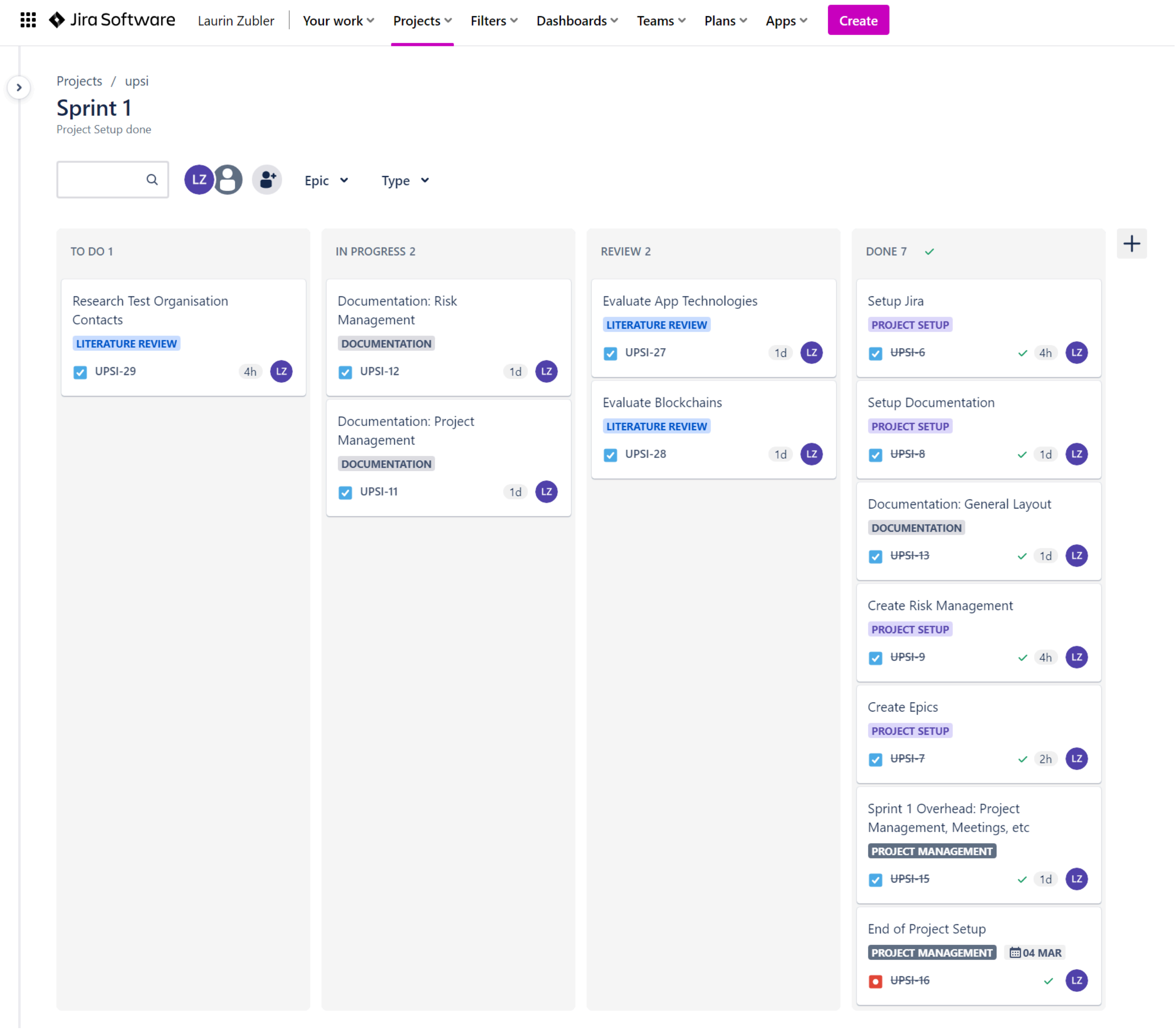


Sprint 1 19 Feb – 4 Mar (9 issues)

Project Setup done

☑ UPSI-6 Setup Jira	PROJECT SETUP	TO DO		4h	LZ
☑ UPSI-7 Create Epics	PROJECT SETUP	TO DO		2h	LZ
☑ UPSI-8 Setup Documentation	PROJECT SETUP	TO DO		1d	LZ
☑ UPSI-9 Create Risk Management	PROJECT SETUP	TO DO		4h	LZ
☑ UPSI-11 Documentation: Project Management	DOCUMENTATION	TO DO		1d	LZ
☑ UPSI-12 Documentation: Risk Management	DOCUMENTATION	TO DO		1d	LZ
☑ UPSI-13 Documentation: General Layout	DOCUMENTATION	TO DO		1d	LZ
☑ UPSI-15 Sprint 1 Overhead: Project Management, Meetings, etc	PROJECT MANAGEMENT	TO DO		1d	LZ
UPSI-16 End of Project Setup	PROJECT MANAGEMENT	TO DO	04 MAR	0m	LZ

Figure 42: Screenshot of the Jira tasks at the start of Sprint 1



Projects / upsi

Sprint 1

Project Setup done

TO DO 1

- Research Test Organisation Contacts — LITERATURE REVIEW — ☑ UPSI-29 — 4h — LZ

IN PROGRESS 2

- Documentation: Risk Management — DOCUMENTATION — ☑ UPSI-12 — 1d — LZ
- Documentation: Project Management — DOCUMENTATION — ☑ UPSI-11 — 1d — LZ

REVIEW 2

- Evaluate App Technologies — LITERATURE REVIEW — ☑ UPSI-27 — 1d — LZ
- Evaluate Blockchains — LITERATURE REVIEW — ☑ UPSI-28 — 1d — LZ

DONE 7 ✓

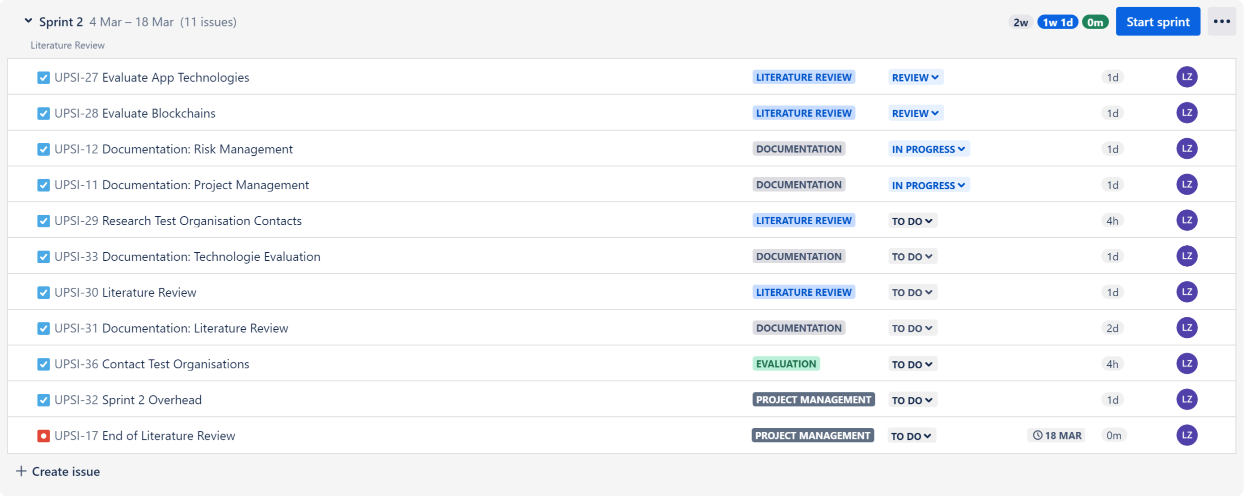
- Setup Jira — PROJECT SETUP — ☑ ~~UPSI-6~~ ✓ 4h — LZ
- Setup Documentation — PROJECT SETUP — ☑ ~~UPSI-8~~ ✓ 1d — LZ
- Documentation: General Layout — DOCUMENTATION — ☑ ~~UPSI-13~~ ✓ 1d — LZ
- Create Risk Management — PROJECT SETUP — ☑ ~~UPSI-9~~ ✓ 4h — LZ
- Create Epics — PROJECT SETUP — ☑ ~~UPSI-7~~ ✓ 2h — LZ
- Sprint 1 Overhead: Project Management, Meetings, etc — PROJECT MANAGEMENT — ☑ ~~UPSI-15~~ ✓ 1d — LZ
- End of Project Setup — PROJECT MANAGEMENT — 04 MAR — ~~UPSI-16~~ ✓ — LZ

Figure 43: Screenshot of the Jira swimlanes at the end of Sprint 1

11.2.8.2 Sprint 2

Period 05.03.2024 - 18.03.2023

Screenshot Jira



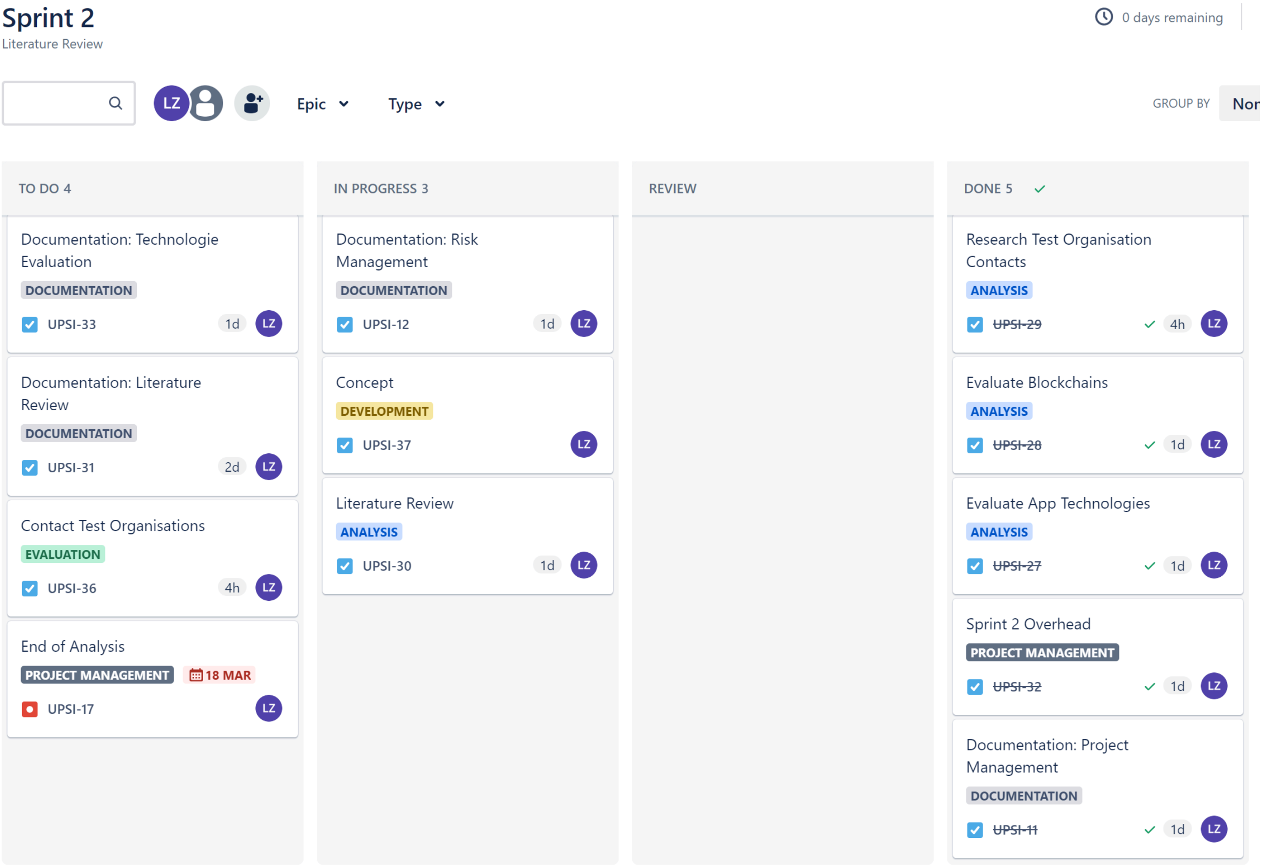
Sprint 2 4 Mar – 18 Mar (11 issues) 2w 1w 1d 0m Start sprint •••

Literature Review

☑	UPSI-27 Evaluate App Technologies	LITERATURE REVIEW	REVIEW		1d	LZ
☑	UPSI-28 Evaluate Blockchains	LITERATURE REVIEW	REVIEW		1d	LZ
☑	UPSI-12 Documentation: Risk Management	DOCUMENTATION	IN PROGRESS		1d	LZ
☑	UPSI-11 Documentation: Project Management	DOCUMENTATION	IN PROGRESS		1d	LZ
☑	UPSI-29 Research Test Organisation Contacts	LITERATURE REVIEW	TO DO		4h	LZ
☑	UPSI-33 Documentation: Technologie Evaluation	DOCUMENTATION	TO DO		1d	LZ
☑	UPSI-30 Literature Review	LITERATURE REVIEW	TO DO		1d	LZ
☑	UPSI-31 Documentation: Literature Review	DOCUMENTATION	TO DO		2d	LZ
☑	UPSI-36 Contact Test Organisations	EVALUATION	TO DO		4h	LZ
☑	UPSI-32 Sprint 2 Overhead	PROJECT MANAGEMENT	TO DO		1d	LZ
■	UPSI-17 End of Literature Review	PROJECT MANAGEMENT	TO DO	18 MAR	0m	LZ

+ Create issue

Figure 44: Screenshot of the Jira tasks at the start of Sprint 2



Sprint 2

Literature Review

0 days remaining

TO DO 4

- Documentation: Technologie Evaluation — DOCUMENTATION — ☑ UPSI-33 1d LZ
- Documentation: Literature Review — DOCUMENTATION — ☑ UPSI-31 2d LZ
- Contact Test Organisations — EVALUATION — ☑ UPSI-36 4h LZ
- End of Analysis — PROJECT MANAGEMENT 18 MAR — ■ UPSI-17 LZ

IN PROGRESS 3

- Documentation: Risk Management — DOCUMENTATION — ☑ UPSI-12 1d LZ
- Concept — DEVELOPMENT — ☑ UPSI-37 LZ
- Literature Review — ANALYSIS — ☑ UPSI-30 1d LZ

REVIEW

DONE 5 ✓

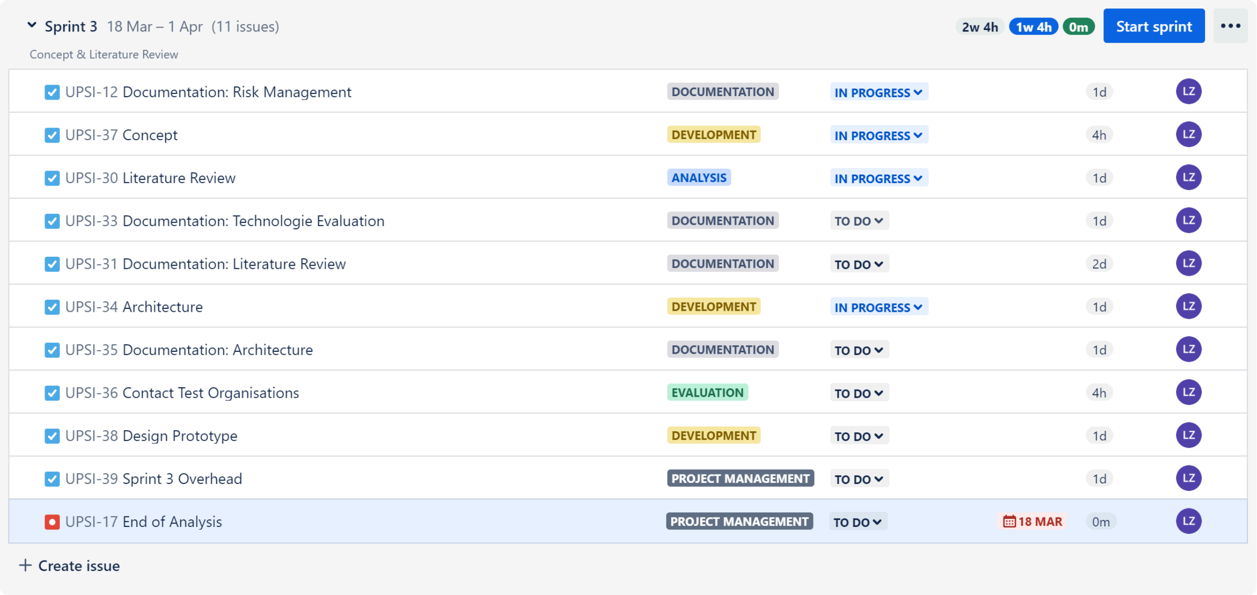
- Research Test Organisation Contacts — ANALYSIS — ☑ ~~UPSI-29~~ ✓ 4h LZ
- Evaluate Blockchains — ANALYSIS — ☑ ~~UPSI-28~~ ✓ 1d LZ
- Evaluate App Technologies — ANALYSIS — ☑ ~~UPSI-27~~ ✓ 1d LZ
- Sprint 2 Overhead — PROJECT MANAGEMENT — ☑ ~~UPSI-32~~ ✓ 1d LZ
- Documentation: Project Management — DOCUMENTATION — ☑ ~~UPSI-11~~ ✓ 1d LZ

Figure 45: Screenshot of the Jira swimlanes at the end of Sprint 2

11.2.8.3 Sprint 3

Period 19.03.2024 - 01.04.2023

Screenshot Jira



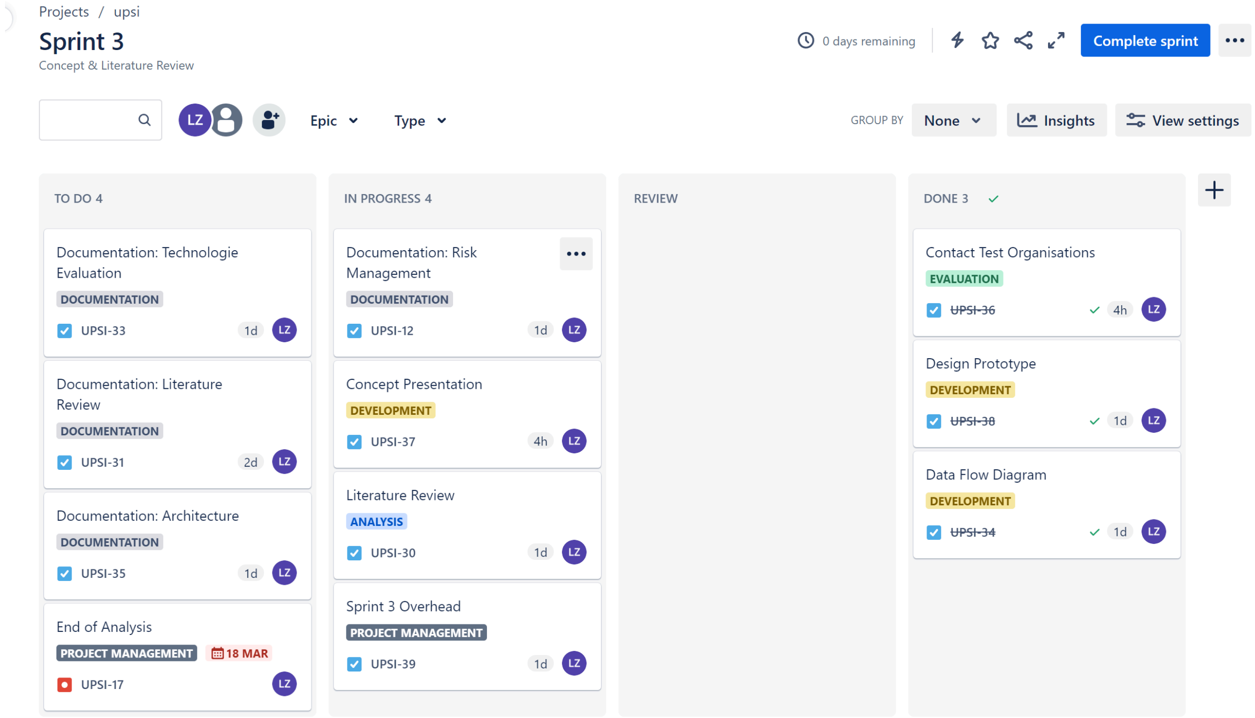
Sprint 3 18 Mar – 1 Apr (11 issues) 2w 4h 1w 4h 0m Start sprint •••

Concept & Literature Review

☑ UPSI-12 Documentation: Risk Management	DOCUMENTATION	IN PROGRESS		1d	LZ
☑ UPSI-37 Concept	DEVELOPMENT	IN PROGRESS		4h	LZ
☑ UPSI-30 Literature Review	ANALYSIS	IN PROGRESS		1d	LZ
☑ UPSI-33 Documentation: Technologie Evaluation	DOCUMENTATION	TO DO		1d	LZ
☑ UPSI-31 Documentation: Literature Review	DOCUMENTATION	TO DO		2d	LZ
☑ UPSI-34 Architecture	DEVELOPMENT	IN PROGRESS		1d	LZ
☑ UPSI-35 Documentation: Architecture	DOCUMENTATION	TO DO		1d	LZ
☑ UPSI-36 Contact Test Organisations	EVALUATION	TO DO		4h	LZ
☑ UPSI-38 Design Prototype	DEVELOPMENT	TO DO		1d	LZ
☑ UPSI-39 Sprint 3 Overhead	PROJECT MANAGEMENT	TO DO		1d	LZ
■ UPSI-17 End of Analysis	PROJECT MANAGEMENT	TO DO	18 MAR	0m	LZ

+ Create issue

Figure 46: Screenshot of the Jira tasks at the start of Sprint 3



Projects / upsi

Sprint 3

Concept & Literature Review

0 days remaining Complete sprint •••

LZ Epic Type GROUP BY None Insights View settings

TO DO 4

- Documentation: Technologie Evaluation — DOCUMENTATION — ☑ UPSI-33 — 1d — LZ
- Documentation: Literature Review — DOCUMENTATION — ☑ UPSI-31 — 2d — LZ
- Documentation: Architecture — DOCUMENTATION — ☑ UPSI-35 — 1d — LZ
- End of Analysis — PROJECT MANAGEMENT — 18 MAR — ■ UPSI-17 — LZ

IN PROGRESS 4

- Documentation: Risk Management — DOCUMENTATION — ☑ UPSI-12 — 1d — LZ
- Concept Presentation — DEVELOPMENT — ☑ UPSI-37 — 4h — LZ
- Literature Review — ANALYSIS — ☑ UPSI-30 — 1d — LZ
- Sprint 3 Overhead — PROJECT MANAGEMENT — ☑ UPSI-39 — 1d — LZ

REVIEW

DONE 3 ✓

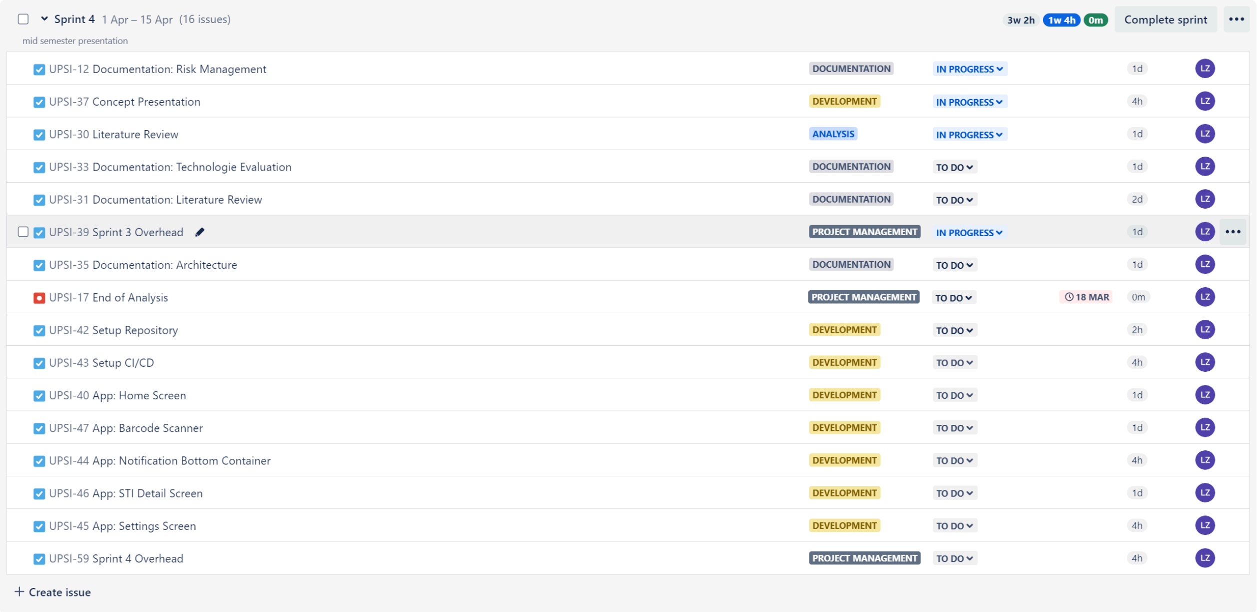
- Contact Test Organisations — EVALUATION — ☑ ~~UPSI-36~~ ✓ 4h — LZ
- Design Prototype — DEVELOPMENT — ☑ ~~UPSI-38~~ ✓ 1d — LZ
- Data Flow Diagram — DEVELOPMENT — ☑ ~~UPSI-34~~ ✓ 1d — LZ

Figure 47: Screenshot of the Jira swimlanes at the end of Sprint 3

11.2.8.4 Sprint 4

Period 02.04.2024 - 15.04.2023

Screenshot Jira

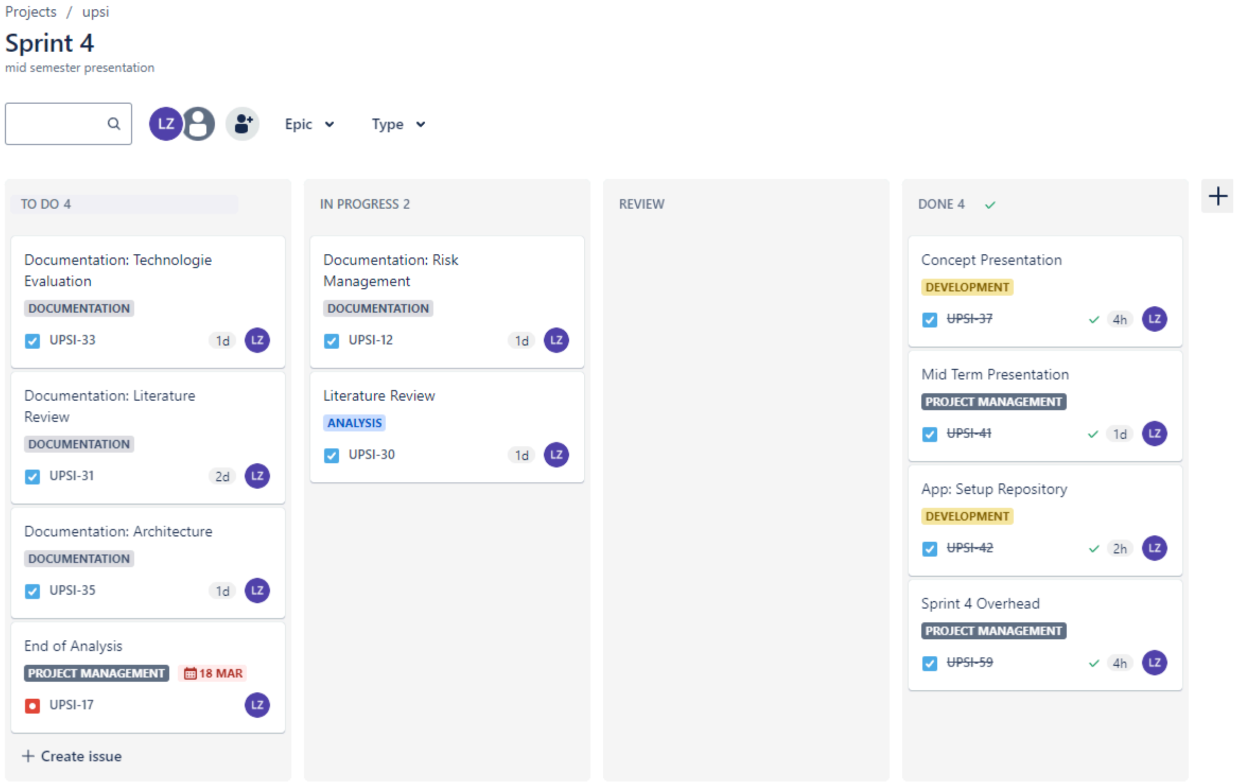


□ Sprint 4 1 Apr – 15 Apr (16 issues) 3w 2h 1w 4h 0m Complete sprint
mid semester presentation

☑ UPSI-12 Documentation: Risk Management	DOCUMENTATION	IN PROGRESS		1d	LZ
☑ UPSI-37 Concept Presentation	DEVELOPMENT	IN PROGRESS		4h	LZ
☑ UPSI-30 Literature Review	ANALYSIS	IN PROGRESS		1d	LZ
☑ UPSI-33 Documentation: Technologie Evaluation	DOCUMENTATION	TO DO		1d	LZ
☑ UPSI-31 Documentation: Literature Review	DOCUMENTATION	TO DO		2d	LZ
☑ UPSI-39 Sprint 3 Overhead	PROJECT MANAGEMENT	IN PROGRESS		1d	LZ
☑ UPSI-35 Documentation: Architecture	DOCUMENTATION	TO DO		1d	LZ
UPSI-17 End of Analysis	PROJECT MANAGEMENT	TO DO	18 MAR	0m	LZ
☑ UPSI-42 Setup Repository	DEVELOPMENT	TO DO		2h	LZ
☑ UPSI-43 Setup CI/CD	DEVELOPMENT	TO DO		4h	LZ
☑ UPSI-40 App: Home Screen	DEVELOPMENT	TO DO		1d	LZ
☑ UPSI-47 App: Barcode Scanner	DEVELOPMENT	TO DO		1d	LZ
☑ UPSI-44 App: Notification Bottom Container	DEVELOPMENT	TO DO		4h	LZ
☑ UPSI-46 App: STI Detail Screen	DEVELOPMENT	TO DO		1d	LZ
☑ UPSI-45 App: Settings Screen	DEVELOPMENT	TO DO		4h	LZ
☑ UPSI-59 Sprint 4 Overhead	PROJECT MANAGEMENT	TO DO		4h	LZ

+ Create issue

Figure 48: Screenshot of the Jira tasks at the start of Sprint 4



Projects / upsi

Sprint 4
mid semester presentation

TO DO 4
- Documentation: Technologie Evaluation — DOCUMENTATION — ☑ UPSI-33 — 1d — LZ
- Documentation: Literature Review — DOCUMENTATION — ☑ UPSI-31 — 2d — LZ
- Documentation: Architecture — DOCUMENTATION — ☑ UPSI-35 — 1d — LZ
- End of Analysis — PROJECT MANAGEMENT — 18 MAR — UPSI-17 — LZ

+ Create issue

IN PROGRESS 2
- Documentation: Risk Management — DOCUMENTATION — ☑ UPSI-12 — 1d — LZ
- Literature Review — ANALYSIS — ☑ UPSI-30 — 1d — LZ

REVIEW

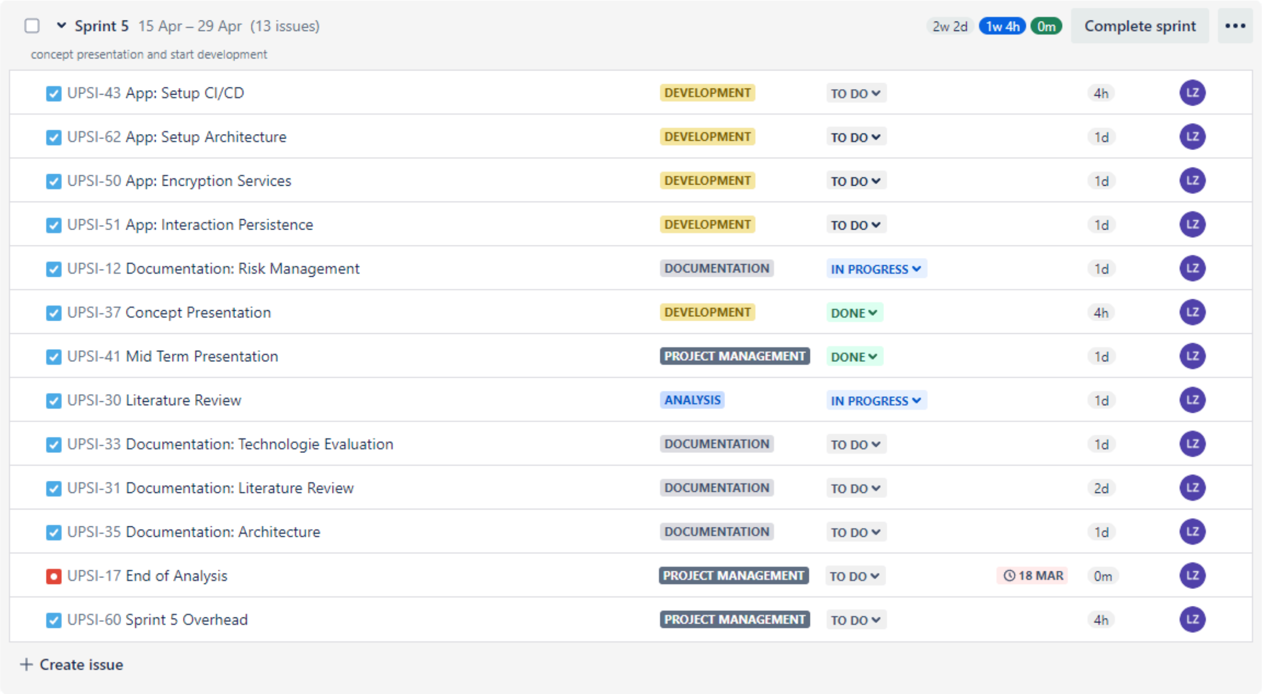
DONE 4 ✓
- Concept Presentation — DEVELOPMENT — ☑ ~~UPSI-37~~ ✓ 4h — LZ
- Mid Term Presentation — PROJECT MANAGEMENT — ☑ ~~UPSI-41~~ ✓ 1d — LZ
- App: Setup Repository — DEVELOPMENT — ☑ ~~UPSI-42~~ ✓ 2h — LZ
- Sprint 4 Overhead — PROJECT MANAGEMENT — ☑ ~~UPSI-59~~ ✓ 4h — LZ

Figure 49: Screenshot of the Jira swimlanes at the end of Sprint 4

11.2.8.5 Sprint 5

Period 16.04.2024 - 29.04.2023

Screenshot Jira



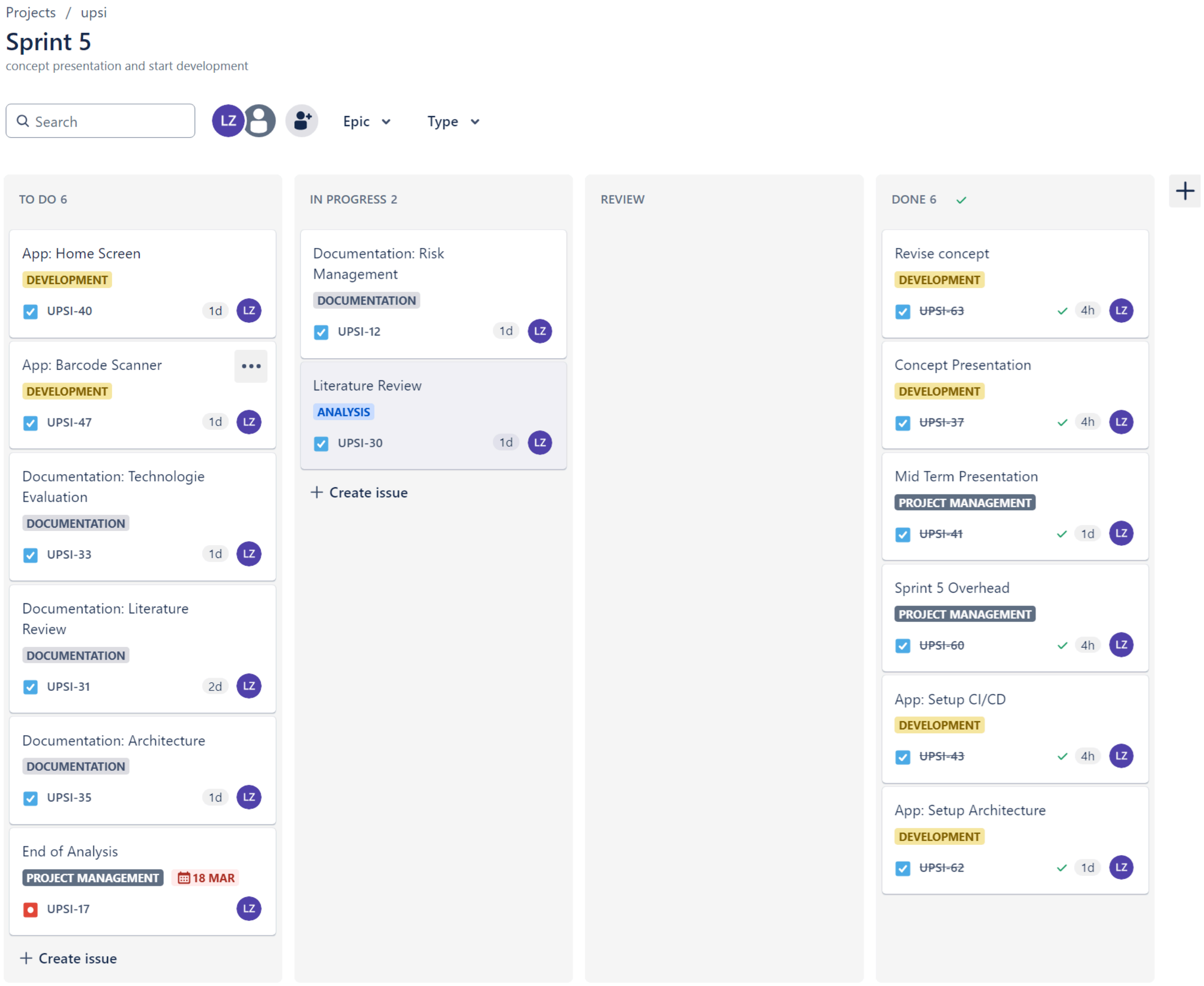
☐ Sprint 5 15 Apr – 29 Apr (13 issues) 2w 2d 1w 4h 0m Complete sprint •••

concept presentation and start development

☑ UPSI-43 App: Setup CI/CD	DEVELOPMENT	TO DO		4h	LZ
☑ UPSI-62 App: Setup Architecture	DEVELOPMENT	TO DO		1d	LZ
☑ UPSI-50 App: Encryption Services	DEVELOPMENT	TO DO		1d	LZ
☑ UPSI-51 App: Interaction Persistence	DEVELOPMENT	TO DO		1d	LZ
☑ UPSI-12 Documentation: Risk Management	DOCUMENTATION	IN PROGRESS		1d	LZ
☑ UPSI-37 Concept Presentation	DEVELOPMENT	DONE		4h	LZ
☑ UPSI-41 Mid Term Presentation	PROJECT MANAGEMENT	DONE		1d	LZ
☑ UPSI-30 Literature Review	ANALYSIS	IN PROGRESS		1d	LZ
☑ UPSI-33 Documentation: Technologie Evaluation	DOCUMENTATION	TO DO		1d	LZ
☑ UPSI-31 Documentation: Literature Review	DOCUMENTATION	TO DO		2d	LZ
☑ UPSI-35 Documentation: Architecture	DOCUMENTATION	TO DO		1d	LZ
● UPSI-17 End of Analysis	PROJECT MANAGEMENT	TO DO	18 MAR	0m	LZ
☑ UPSI-60 Sprint 5 Overhead	PROJECT MANAGEMENT	TO DO		4h	LZ

+ Create issue

Figure 50: Screenshot of the Jira tasks at the start of Sprint 5



Projects / upsi

Sprint 5

concept presentation and start development

Search LZ Epic Type

TO DO 6

App: Home Screen
DEVELOPMENT
☑ UPSI-40 1d LZ

App: Barcode Scanner
DEVELOPMENT
☑ UPSI-47 1d LZ

Documentation: Technologie Evaluation
DOCUMENTATION
☑ UPSI-33 1d LZ

Documentation: Literature Review
DOCUMENTATION
☑ UPSI-31 2d LZ

Documentation: Architecture
DOCUMENTATION
☑ UPSI-35 1d LZ

End of Analysis
PROJECT MANAGEMENT 18 MAR
🟥 UPSI-17 LZ

+ Create issue

IN PROGRESS 2

Documentation: Risk Management
DOCUMENTATION
☑ UPSI-12 1d LZ

Literature Review
ANALYSIS
☑ UPSI-30 1d LZ

+ Create issue

REVIEW

DONE 6 ✓

Revise concept
DEVELOPMENT
☑ ~~UPSI-63~~ ✓ 4h LZ

Concept Presentation
DEVELOPMENT
☑ ~~UPSI-37~~ ✓ 4h LZ

Mid Term Presentation
PROJECT MANAGEMENT
☑ ~~UPSI-41~~ ✓ 1d LZ

Sprint 5 Overhead
PROJECT MANAGEMENT
☑ ~~UPSI-60~~ ✓ 4h LZ

App: Setup CI/CD
DEVELOPMENT
☑ ~~UPSI-43~~ ✓ 4h LZ

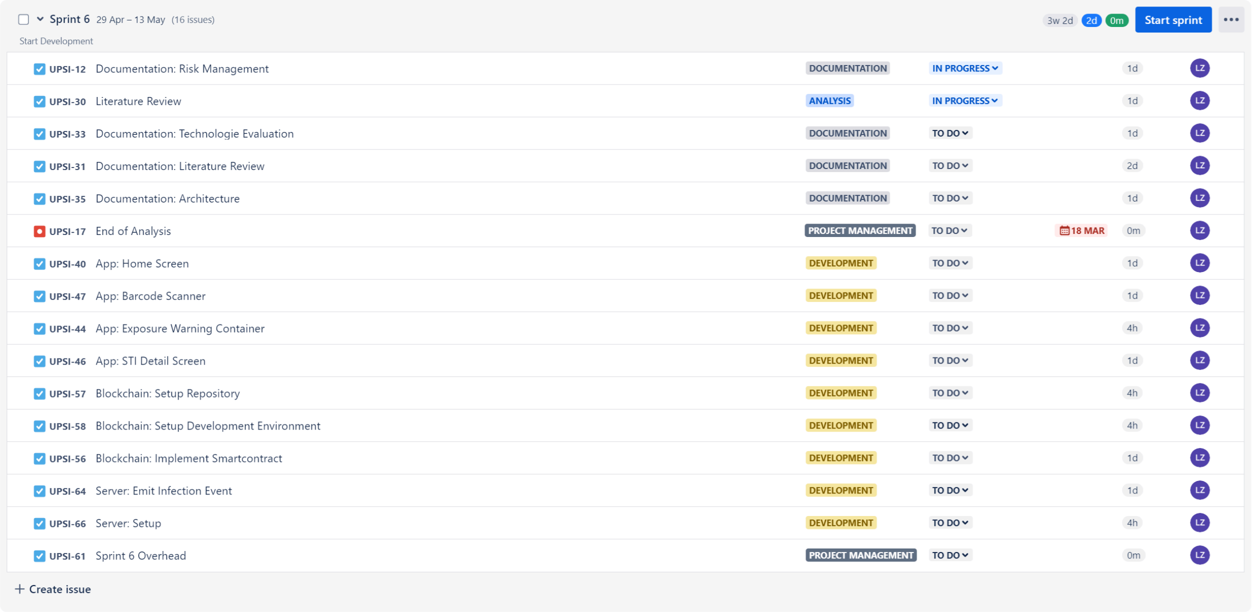
App: Setup Architecture
DEVELOPMENT
☑ ~~UPSI-62~~ ✓ 1d LZ

Figure 51: Screenshot of the Jira swimlanes at the end of Sprint 5

11.2.8.6 Sprint 6

Period 30.04.2024 - 13.05.2023

Screenshot Jira



☐ Sprint 6 29 Apr – 13 May (16 issues)

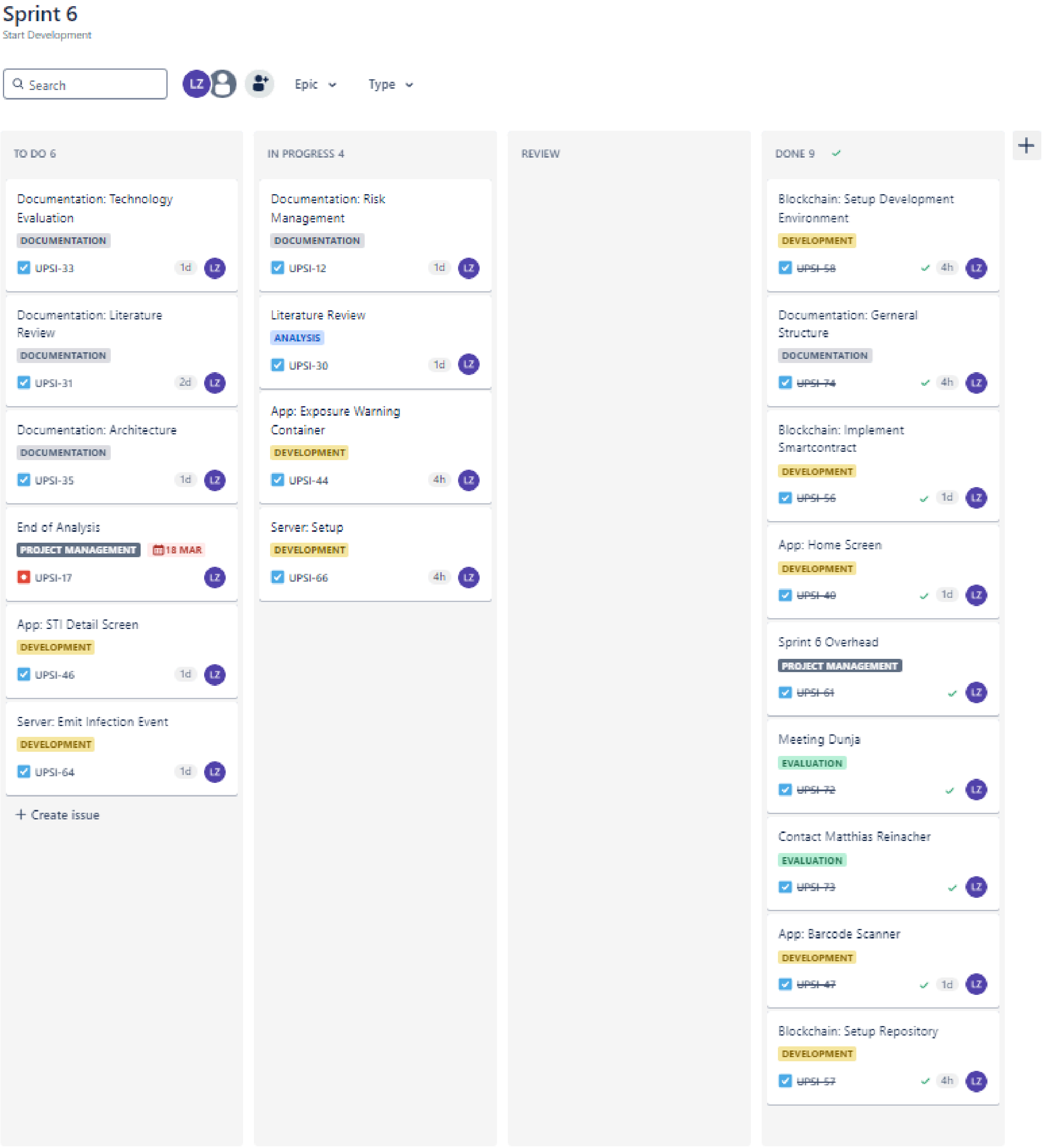
3w 2d 2d 0m Start sprint •••

Start Development

☑	UPSI-12	Documentation: Risk Management	DOCUMENTATION	IN PROGRESS		1d	LZ
☑	UPSI-30	Literature Review	ANALYSIS	IN PROGRESS		1d	LZ
☑	UPSI-33	Documentation: Technologie Evaluation	DOCUMENTATION	TO DO		1d	LZ
☑	UPSI-31	Documentation: Literature Review	DOCUMENTATION	TO DO		2d	LZ
☑	UPSI-35	Documentation: Architecture	DOCUMENTATION	TO DO		1d	LZ
🟥	UPSI-17	End of Analysis	PROJECT MANAGEMENT	TO DO	18 MAR	0m	LZ
☑	UPSI-40	App: Home Screen	DEVELOPMENT	TO DO		1d	LZ
☑	UPSI-47	App: Barcode Scanner	DEVELOPMENT	TO DO		1d	LZ
☑	UPSI-44	App: Exposure Warning Container	DEVELOPMENT	TO DO		4h	LZ
☑	UPSI-46	App: STI Detail Screen	DEVELOPMENT	TO DO		1d	LZ
☑	UPSI-57	Blockchain: Setup Repository	DEVELOPMENT	TO DO		4h	LZ
☑	UPSI-58	Blockchain: Setup Development Environment	DEVELOPMENT	TO DO		4h	LZ
☑	UPSI-56	Blockchain: Implement Smartcontract	DEVELOPMENT	TO DO		1d	LZ
☑	UPSI-64	Server: Emit Infection Event	DEVELOPMENT	TO DO		1d	LZ
☑	UPSI-66	Server: Setup	DEVELOPMENT	TO DO		4h	LZ
☑	UPSI-61	Sprint 6 Overhead	PROJECT MANAGEMENT	TO DO		0m	LZ

+ Create issue

Figure 52: Screenshot of the Jira tasks at the start of Sprint 6



Sprint 6
Start Development

Search · LZ · Epic · Type

TO DO 6

Documentation: Technology Evaluation
DOCUMENTATION
☑ UPSI-33 · 1d · LZ

Documentation: Literature Review
DOCUMENTATION
☑ UPSI-31 · 2d · LZ

Documentation: Architecture
DOCUMENTATION
☑ UPSI-35 · 1d · LZ

End of Analysis
PROJECT MANAGEMENT · 18 MAR
UPSI-17 · LZ

App: STI Detail Screen
DEVELOPMENT
☑ UPSI-46 · 1d · LZ

Server: Emit Infection Event
DEVELOPMENT
☑ UPSI-64 · 1d · LZ

+ Create issue

IN PROGRESS 4

Documentation: Risk Management
DOCUMENTATION
☑ UPSI-12 · 1d · LZ

Literature Review
ANALYSIS
☑ UPSI-30 · 1d · LZ

App: Exposure Warning Container
DEVELOPMENT
☑ UPSI-44 · 4h · LZ

Server: Setup
DEVELOPMENT
☑ UPSI-66 · 4h · LZ

REVIEW

DONE 9 ✓

Blockchain: Setup Development Environment
DEVELOPMENT
☑ ~~UPSI-58~~ ✓ 4h · LZ

Documentation: Gerneral Structure
DOCUMENTATION
☑ ~~UPSI-74~~ ✓ 4h · LZ

Blockchain: Implement Smartcontract
DEVELOPMENT
☑ ~~UPSI-56~~ ✓ 1d · LZ

App: Home Screen
DEVELOPMENT
☑ ~~UPSI-40~~ ✓ 1d · LZ

Sprint 6 Overhead
PROJECT MANAGEMENT
☑ ~~UPSI-61~~ ✓ LZ

Meeting Dunja
EVALUATION
☑ ~~UPSI-72~~ ✓ LZ

Contact Matthias Reinacher
EVALUATION
☑ ~~UPSI-73~~ ✓ LZ

App: Barcode Scanner
DEVELOPMENT
☑ ~~UPSI-47~~ ✓ 1d · LZ

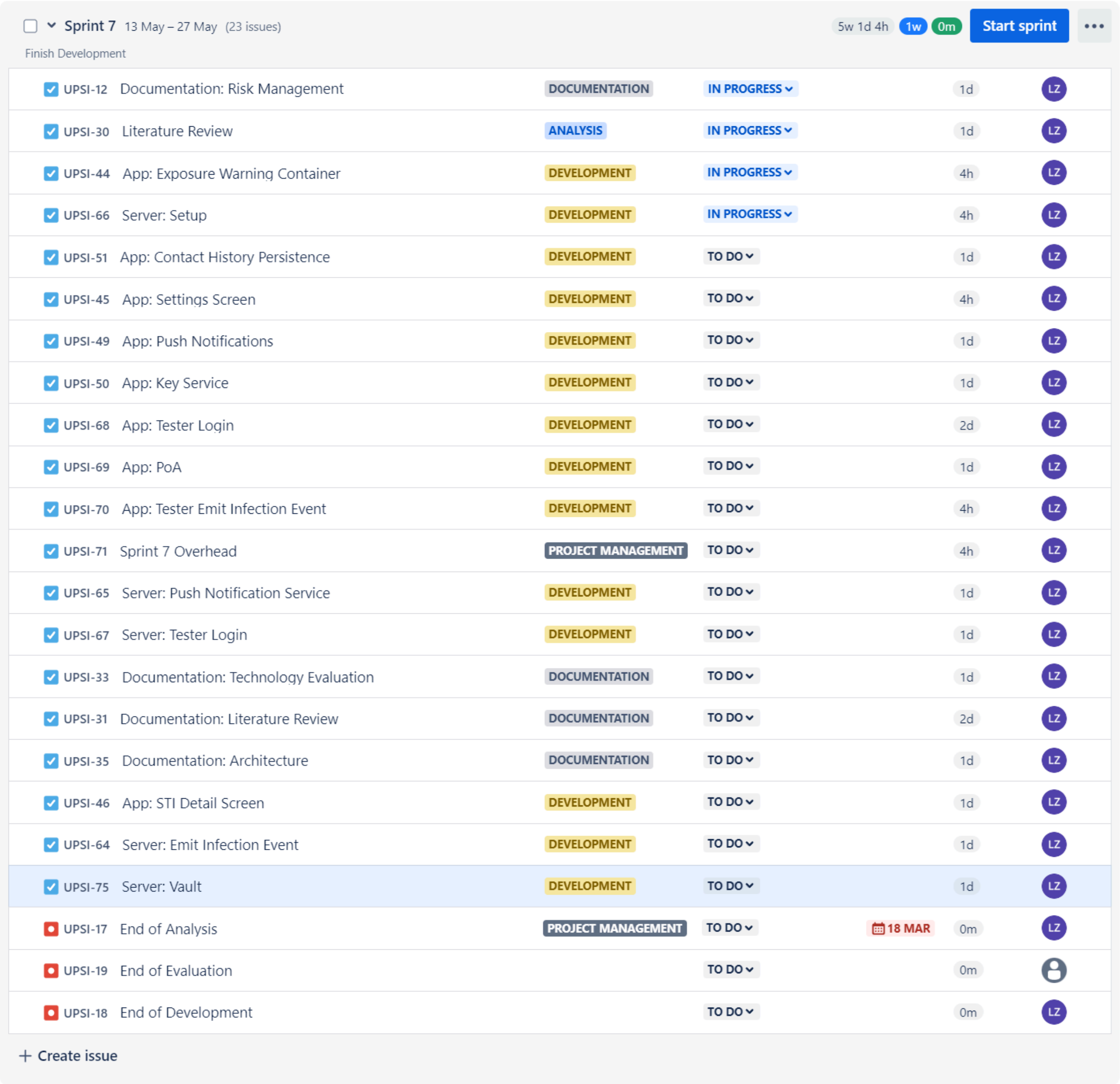
Blockchain: Setup Repository
DEVELOPMENT
☑ ~~UPSI-57~~ ✓ 4h · LZ

Figure 53: Screenshot of the Jira *swimlanes at the end of Sprint 6*

11.2.8.7 Sprint 7

Period 14.05.2024 - 27.05.2023

Screenshot Jira



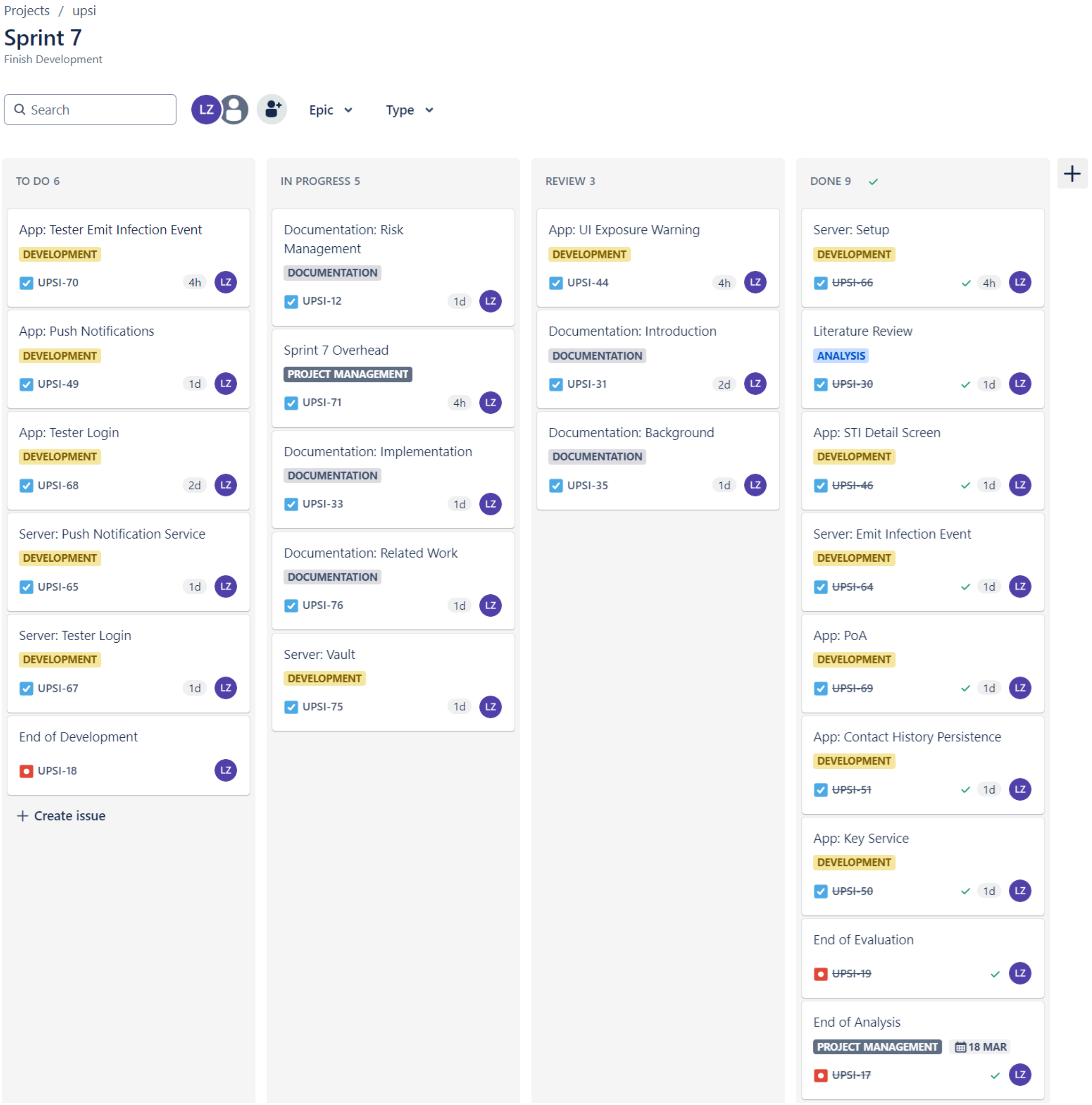
□ ⌄ Sprint 7 13 May – 27 May (23 issues) 5w 1d 4h 1w 0m Start sprint •••

Finish Development

☑	UPSI-12	Documentation: Risk Management	DOCUMENTATION	IN PROGRESS ⌄		1d	LZ
☑	UPSI-30	Literature Review	ANALYSIS	IN PROGRESS ⌄		1d	LZ
☑	UPSI-44	App: Exposure Warning Container	DEVELOPMENT	IN PROGRESS ⌄		4h	LZ
☑	UPSI-66	Server: Setup	DEVELOPMENT	IN PROGRESS ⌄		4h	LZ
☑	UPSI-51	App: Contact History Persistence	DEVELOPMENT	TO DO ⌄		1d	LZ
☑	UPSI-45	App: Settings Screen	DEVELOPMENT	TO DO ⌄		4h	LZ
☑	UPSI-49	App: Push Notifications	DEVELOPMENT	TO DO ⌄		1d	LZ
☑	UPSI-50	App: Key Service	DEVELOPMENT	TO DO ⌄		1d	LZ
☑	UPSI-68	App: Tester Login	DEVELOPMENT	TO DO ⌄		2d	LZ
☑	UPSI-69	App: PoA	DEVELOPMENT	TO DO ⌄		1d	LZ
☑	UPSI-70	App: Tester Emit Infection Event	DEVELOPMENT	TO DO ⌄		4h	LZ
☑	UPSI-71	Sprint 7 Overhead	PROJECT MANAGEMENT	TO DO ⌄		4h	LZ
☑	UPSI-65	Server: Push Notification Service	DEVELOPMENT	TO DO ⌄		1d	LZ
☑	UPSI-67	Server: Tester Login	DEVELOPMENT	TO DO ⌄		1d	LZ
☑	UPSI-33	Documentation: Technology Evaluation	DOCUMENTATION	TO DO ⌄		1d	LZ
☑	UPSI-31	Documentation: Literature Review	DOCUMENTATION	TO DO ⌄		2d	LZ
☑	UPSI-35	Documentation: Architecture	DOCUMENTATION	TO DO ⌄		1d	LZ
☑	UPSI-46	App: STI Detail Screen	DEVELOPMENT	TO DO ⌄		1d	LZ
☑	UPSI-64	Server: Emit Infection Event	DEVELOPMENT	TO DO ⌄		1d	LZ
☑	UPSI-75	Server: Vault	DEVELOPMENT	TO DO ⌄		1d	LZ
■	UPSI-17	End of Analysis	PROJECT MANAGEMENT	TO DO ⌄	18 MAR	0m	LZ
■	UPSI-19	End of Evaluation		TO DO ⌄		0m	
■	UPSI-18	End of Development		TO DO ⌄		0m	LZ

+ Create issue

Figure 54: Screenshot of the Jira tasks at the start of Sprint 7



Projects / upsi

Sprint 7

Finish Development

Search

LZ

Epic ⌄ Type ⌄

TO DO 6

App: Tester Emit Infection Event
DEVELOPMENT
☑ UPSI-70 4h LZ

App: Push Notifications
DEVELOPMENT
☑ UPSI-49 1d LZ

App: Tester Login
DEVELOPMENT
☑ UPSI-68 2d LZ

Server: Push Notification Service
DEVELOPMENT
☑ UPSI-65 1d LZ

Server: Tester Login
DEVELOPMENT
☑ UPSI-67 1d LZ

End of Development
■ UPSI-18 LZ

+ Create issue

IN PROGRESS 5

Documentation: Risk Management
DOCUMENTATION
☑ UPSI-12 1d LZ

Sprint 7 Overhead
PROJECT MANAGEMENT
☑ UPSI-71 4h LZ

Documentation: Implementation
DOCUMENTATION
☑ UPSI-33 1d LZ

Documentation: Related Work
DOCUMENTATION
☑ UPSI-76 1d LZ

Server: Vault
DEVELOPMENT
☑ UPSI-75 1d LZ

REVIEW 3

App: UI Exposure Warning
DEVELOPMENT
☑ UPSI-44 4h LZ

Documentation: Introduction
DOCUMENTATION
☑ UPSI-31 2d LZ

Documentation: Background
DOCUMENTATION
☑ UPSI-35 1d LZ

DONE 9 ✓

Server: Setup
DEVELOPMENT
☑ ~~UPSI-66~~ ✓ 4h LZ

Literature Review
ANALYSIS
☑ ~~UPSI-30~~ ✓ 1d LZ

App: STI Detail Screen
DEVELOPMENT
☑ ~~UPSI-46~~ ✓ 1d LZ

Server: Emit Infection Event
DEVELOPMENT
☑ ~~UPSI-64~~ ✓ 1d LZ

App: PoA
DEVELOPMENT
☑ ~~UPSI-69~~ ✓ 1d LZ

App: Contact History Persistence
DEVELOPMENT
☑ ~~UPSI-51~~ ✓ 1d LZ

App: Key Service
DEVELOPMENT
☑ ~~UPSI-50~~ ✓ 1d LZ

End of Evaluation
■ ~~UPSI-19~~ ✓ LZ

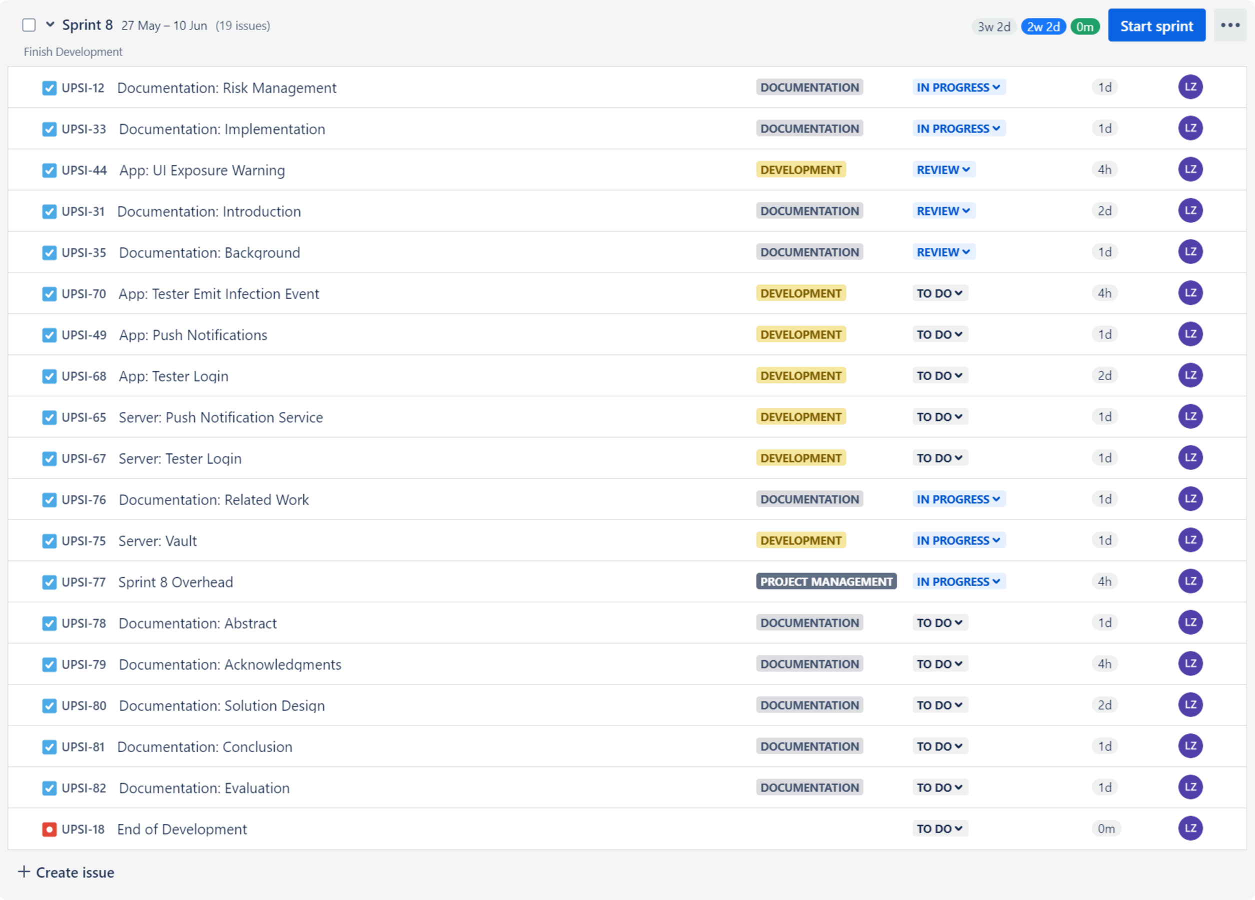
End of Analysis
PROJECT MANAGEMENT 18 MAR
■ ~~UPSI-17~~ ✓ LZ

Figure 55: Screenshot of the Jira swimlanes at the end of Sprint 7

11.2.8.8 Sprint 8

Period 28.05.2024 - 14.06.2023

Screenshot Jira



☐ Sprint 8 27 May – 10 Jun (19 issues)

Finish Development

UPSI-12	Documentation: Risk Management	DOCUMENTATION	IN PROGRESS	1d	LZ
UPSI-33	Documentation: Implementation	DOCUMENTATION	IN PROGRESS	1d	LZ
UPSI-44	App: UI Exposure Warning	DEVELOPMENT	REVIEW	4h	LZ
UPSI-31	Documentation: Introduction	DOCUMENTATION	REVIEW	2d	LZ
UPSI-35	Documentation: Background	DOCUMENTATION	REVIEW	1d	LZ
UPSI-70	App: Tester Emit Infection Event	DEVELOPMENT	TO DO	4h	LZ
UPSI-49	App: Push Notifications	DEVELOPMENT	TO DO	1d	LZ
UPSI-68	App: Tester Login	DEVELOPMENT	TO DO	2d	LZ
UPSI-65	Server: Push Notification Service	DEVELOPMENT	TO DO	1d	LZ
UPSI-67	Server: Tester Login	DEVELOPMENT	TO DO	1d	LZ
UPSI-76	Documentation: Related Work	DOCUMENTATION	IN PROGRESS	1d	LZ
UPSI-75	Server: Vault	DEVELOPMENT	IN PROGRESS	1d	LZ
UPSI-77	Sprint 8 Overhead	PROJECT MANAGEMENT	IN PROGRESS	4h	LZ
UPSI-78	Documentation: Abstract	DOCUMENTATION	TO DO	1d	LZ
UPSI-79	Documentation: Acknowledgments	DOCUMENTATION	TO DO	4h	LZ
UPSI-80	Documentation: Solution Design	DOCUMENTATION	TO DO	2d	LZ
UPSI-81	Documentation: Conclusion	DOCUMENTATION	TO DO	1d	LZ
UPSI-82	Documentation: Evaluation	DOCUMENTATION	TO DO	1d	LZ
UPSI-18	End of Development		TO DO	0m	LZ

Figure 56: Screenshot of the Jira tasks at the start of Sprint 8

11.3 Risk Management

Introduction

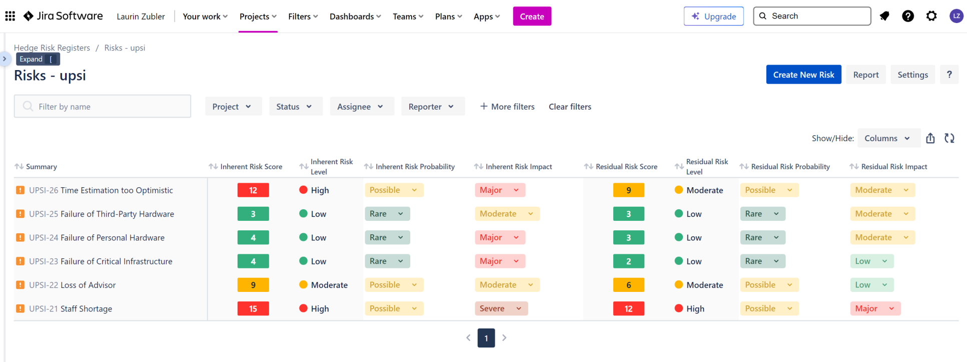
This section highlights the strategies for dealing with risks during the bachelor's theses. The risk management involved the identification, analysis, and mitigation of possible risks.

11.3.2 Risk Analysis

Introduction

In the risk analysis process, identified risks are assigned a probability of occurrence and an impact level should it occur. These values are used to calculate a risk score, which helps to classify the risk as low, moderate, or high according to the risk matrix. This classification is done twice: firstly, to determine the inherent risk level, which is the risk level before any countermeasures are implemented, and secondly, to assess the residual risk level, which is the risk level after the implementation of countermeasures.

Figure 57 presents a screenshot of the *Hedge: Risk Management, Risk Register & Risk Matrix for Jira*, illustrating how the risks were organized.



Risks - upsi

Summary	Inherent Risk Score	Inherent Risk Level	Inherent Risk Probability	Inherent Risk Impact	Residual Risk Score	Residual Risk Level	Residual Risk Probability	Residual Risk Impact
UPSI-26 Time Estimation too Optimistic	12	● High	Possible	Major	9	● Moderate	Possible	Moderate
UPSI-25 Failure of Third-Party Hardware	3	● Low	Rare	Moderate	3	● Low	Rare	Moderate
UPSI-24 Failure of Personal Hardware	4	● Low	Rare	Major	3	● Low	Rare	Moderate
UPSI-23 Failure of Critical Infrastructure	4	● Low	Rare	Major	2	● Low	Rare	Low
UPSI-22 Loss of Advisor	9	● Moderate	Possible	Moderate	6	● Moderate	Possible	Low
UPSI-21 Staff Shortage	15	● High	Possible	Severe	12	● High	Possible	Major

Figure 57: Screenshot of identified risks in Jira

Risk Metrics

Table 9 illustrates the specific values assigned to each risk probability and impact, which are used in calculating the risk score used for the classification of the risk.

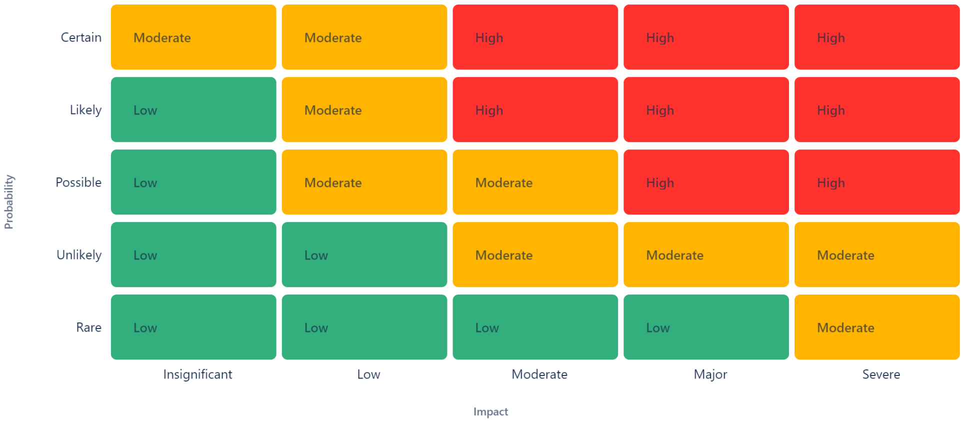
Probability	Impact	Value
Rare	Insignificant	1
Unlikely	Low	2
Possible	Moderate	3
Likely	Major	4
Certain	Severe	5

Table 9: Risk matrix to calculate the risk score

Risks Matrix

Using the probability and impact each risk is classified as *Low, Moderate* or *High*.

Figure 58 shows how the risk classes are distributed.



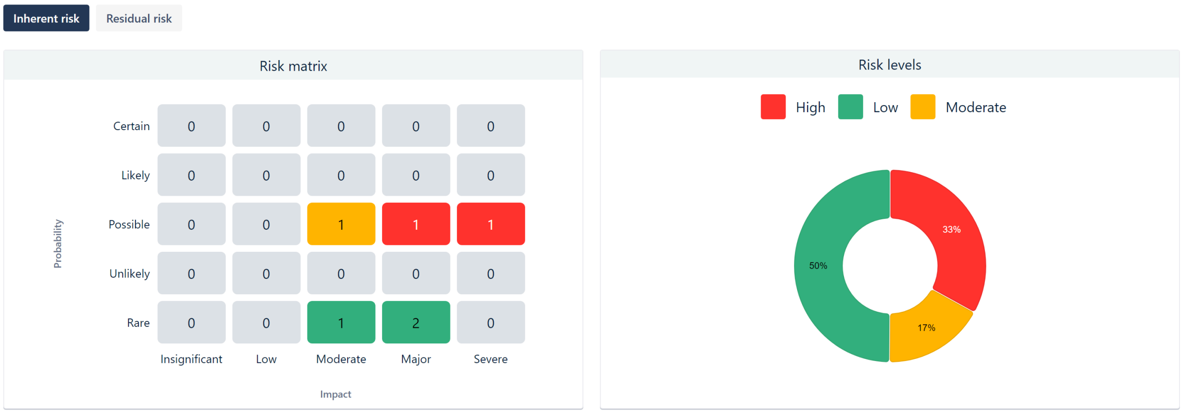
Probability	Insignificant	Low	Moderate	Major	Severe
Certain	Moderate	Moderate	High	High	High
Likely	Low	Moderate	High	High	High
Possible	Low	Moderate	Moderate	High	High
Unlikely	Low	Low	Moderate	Moderate	Moderate
Rare	Low	Low	Low	Low	Moderate

Figure 58: Risk matrix class distribution

Inherent Risk Report

The inherent risks report is providing a visual representation of the potential risks identified, before any mitigating actions are taken.

Figure 59 presents a screenshot from *Jira* of the inherent risks report, showcasing how each risk is categorized and assessed.



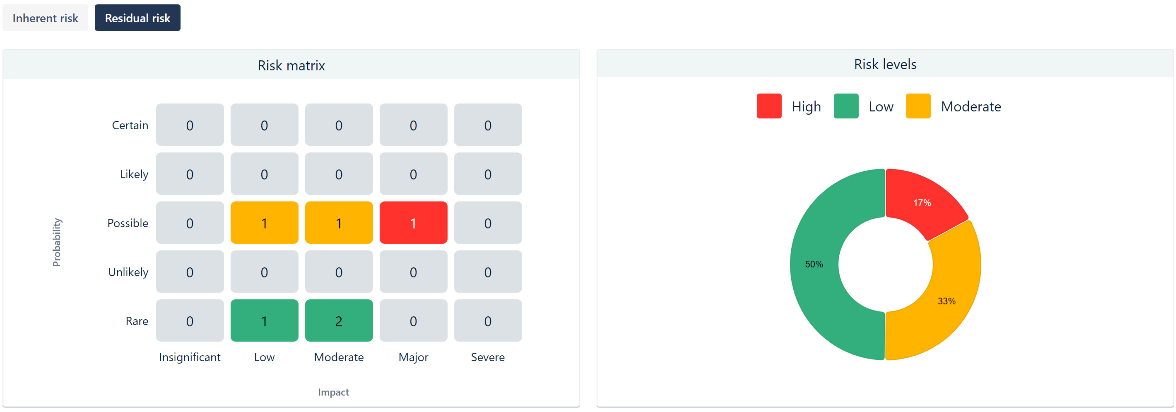
Probability	Insignificant	Low	Moderate	Major	Severe
Certain	0	0	0	0	0
Likely	0	0	0	0	0
Possible	0	0	1	1	1
Unlikely	0	0	0	0	0
Rare	0	0	1	2	0

Figure 59: Screenshot of inherent risk report in Jira

Residual Risk Report

The residual risks report is providing a visual representation of the potential risks identified, after the implementation of countermeasures.

Figure 59 presents a screenshot from *Jira* of the residual risks report, showcasing how each risk is categorized.



Probability	Insignificant	Low	Moderate	Major	Severe
Certain	0	0	0	0	0
Likely	0	0	0	0	0
Possible	0	1	1	1	0
Unlikely	0	0	0	0	0
Rare	0	1	2	0	0

Impact

Figure 60: Screenshot of residual risk report in Jira

Results

In the risk analysis, six distinct risks were identified. Of these, two were classified as high-risk, one as moderate-risk, and the remaining three as low-risk. Following the implementation of countermeasures, there was a notable reduction in the overall risk levels. While the probability of these risks occurring remained unchanged, the impact of their occurrence was successfully mitigated.

11.3.3 Risks

Introduction In this section, the identified risks are further detailed.

11.3.3.2 Staff Shortage

Description The author of this thesis is no longer able to work due to illness, accidents or other unforeseen events.

Counter-measures The project documentation is constantly updated and working hours are documented in order to identify the project status at any time and to be able to resume work after an absence.

Actions on Risk Occurrence If the loss of work is only for a short duration a request to postpone the final deadline can be submitted. Otherwise the thesis must be canceled or postponed to another semester.

Risk assessment

Type	Probabiliy	Impact	Level
Inherent	Possible	Severe	High
Residual	Possible	Major	High

Figure 61: Risk assessment for Staff Shortage

11.3.3.3 Loss of Advisor

Description The advisor of this thesis is no longer able to advise due to illness, accidents or other unforeseen events.

Counter-measures The project documentation is constantly updated and working hours are documented in order to identify the project status at any time and to be able to resume advisement after an absence.

Actions on Risk Occurrence If the loss of advisor is only for a short duration the thesis can be continued independently. Does the absence last longer and no representative advisor takes over, a complaint can be submitted to the academic advisor, the course director or the university management.

Risk assessment

Type	Probabiliy	Impact	Level
Inherent	Possible	Moderate	Moderate
Residual	Possible	Low	Moderate

Figure 62: Risk assessment for Loss of Advisor

11.3.3.4 Failure of Critical Infrastructure

Description The infrastructure used for writing and implementing this theses, mostly cloud services, can no longer be used.

Counter-measures The probability of this occurring is very low and therefore no explicit countermeasures are taken.

Actions on Risk Occurrence Depending on the failing infrastructure, a solution must be found spontaneously.

Risk assessment

Type	Probabiliy	Impact	Level
Inherent	Rare	Major	Low
Residual	Rare	Low	Low

Figure 63: Risk assessment for Failure of Critical Infrastructure

11.3.3.5 Failure of Personal Hardware

Description The author's hardware for writing and implementing this thesis breaks or can no longer be used for other reasons.

Counter-measures The probability of this occurring is very low and therefore no explicit countermeasures are taken. The author has household contents insurance which can compensate for damages.

Actions on Risk Occurrence Depending on the failing hardware, a solution must be found spontaneously. As backup the university has hardware that can be used.

Risk assessment

Type	Probabiliy	Impact	Level
Inherent	Rare	Major	Low
Residual	Rare	Moderate	Low

Figure 64: Risk assessment for Failure of Personal Hardware

11.3.3.6 Failure of Third-Party Hardware

Description Third-Party hardware for writing and implementing this thesis breaks or can no longer be used for other reasons.

Counter-measures The probability of this occurring is very low and therefore no explicit countermeasures are taken. The author has liability insurance which can compensate for damages.

Actions on Risk Occurrence Depending on the failing hardware, a solution must be found spontaneously.

Risk assessment

Type	Probabiliy	Impact	Level
Inherent	Rare	Moderate	Low
Residual	Rare	Moderate	Low

Figure 65: Risk assessment for Failure of Third-Party Hardware

11.3.3.7 Time Estimation too Optimistic

Description Tasks take longer than expected and the planned goal cannot be reached.

Counter-measures The working hours are documented in order to identify the the working effort, to explain if something cannot be done.

Actions on Risk Occurrence Low priority features are not implemented.

Risk assessment

Type	Probabiliy	Impact	Level
Inherent	Possible	Major	High
Residual	Possible	Moderate	Moderate

Figure 66: Risk assessment for Time Estimation too Optimistic

11.4 Quality Assurance

Introduction In this section, the various actions undertaken to ensure good quality and safety are described.

11.4.2 Project Documentation

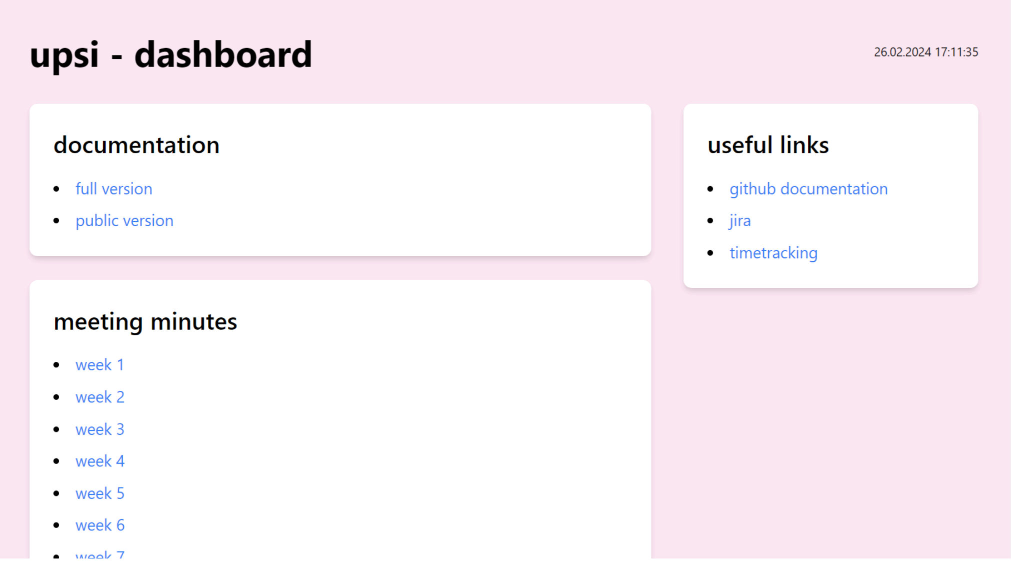
Typst The project documentation was written using Typst[81].

GitHub For the validation of versioning and backups of the document, GitHub[82] was utilized. This platform ensured that all changes and iterations of the project documentation were systematically tracked and securely stored. To further streamline the process, a CI/CD (Continuous Integration/Continuous Deployment) pipeline was implemented. This setup facilitated the automated building of the document.

In the event of any failures or issues in the pipeline, a notification system was established using a Microsoft Teams channel. This integration allowed for immediate alerts from GitHub whenever the pipeline encountered problems, ensuring prompt attention and resolution.

Dashboard For the distribution of the project documentation, meeting minutes, and essential links, GitHub Pages was employed to create a online project dashboard. This dashboard provided a centralized and accessible location for all project-related materials. This ensured that the most recent versions of the documents were always available, particularly beneficial for the advisor, who could access the latest work at any time.

Figure 67 is showing a screenshot of the project dashboard.



upsi - dashboard

26.02.2024 17:11:35

documentation

- full version
- public version

useful links

- github documentation
- jira
- timetracking

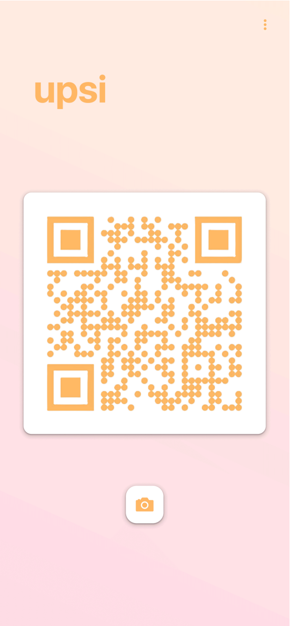
meeting minutes

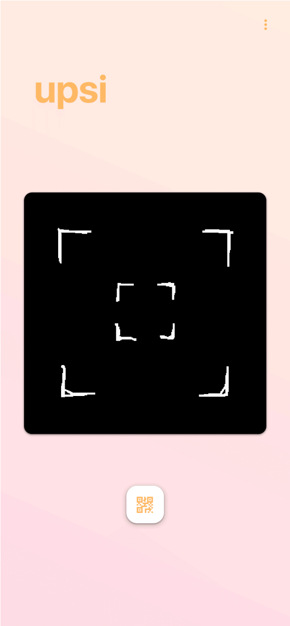
- week 1
- week 2
- week 3
- week 4
- week 5
- week 6
- week 7

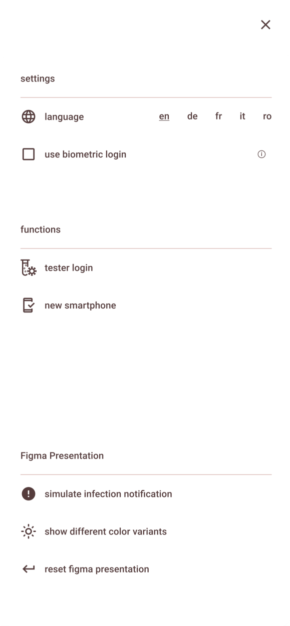
Figure 67: Screenshot of the project dashboard

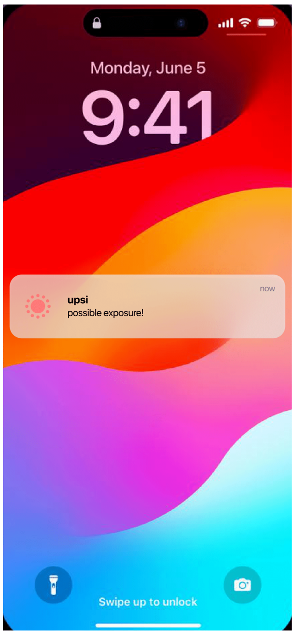
[81] https://typst.app/
[82] https://github.com/

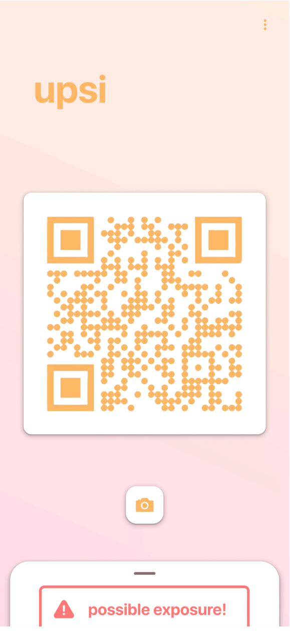
11.5 Figma Design Prototype Screens

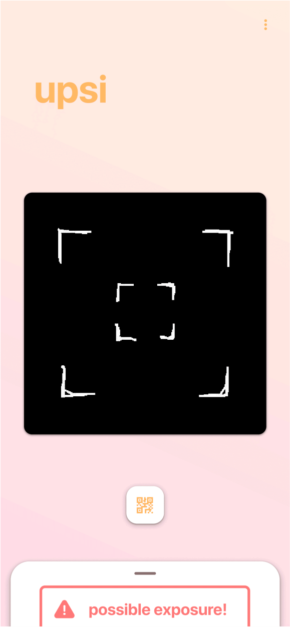


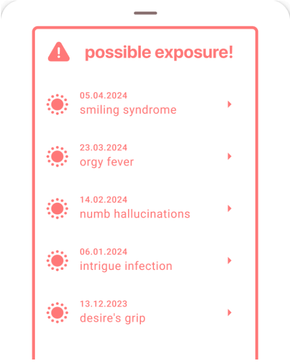











possible exposure!

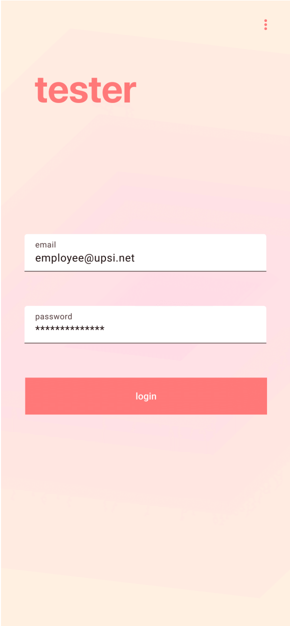
05.04.2024
smiling syndrome

23.03.2024
orgy fever

14.02.2024
numb hallucinations

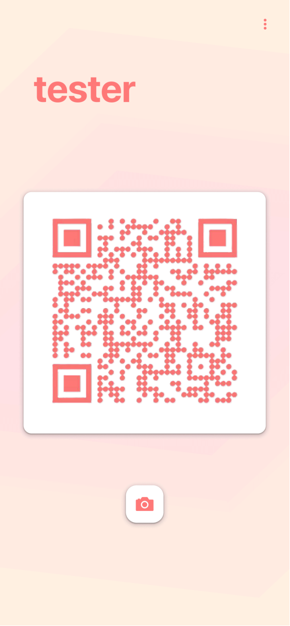
06.01.2024
intrigue infection

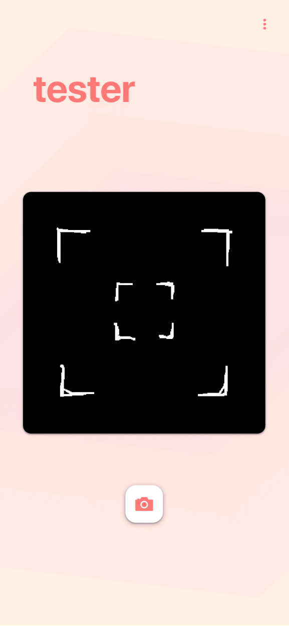
13.12.2023
desire's grip

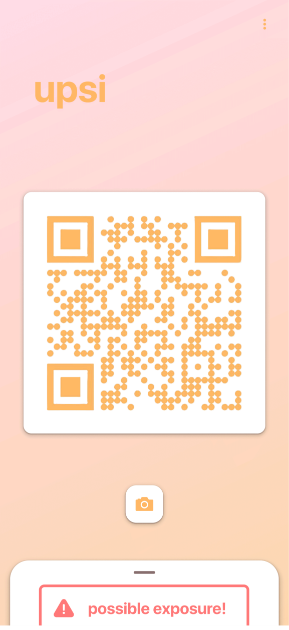

tester

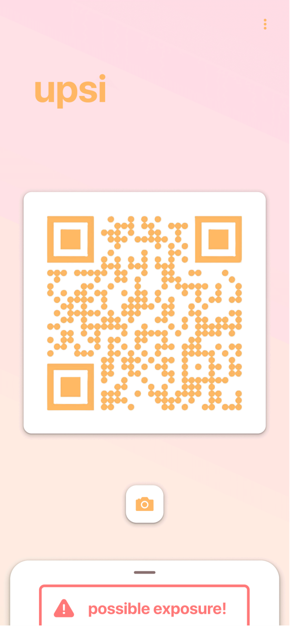
email
employee@upsi.net

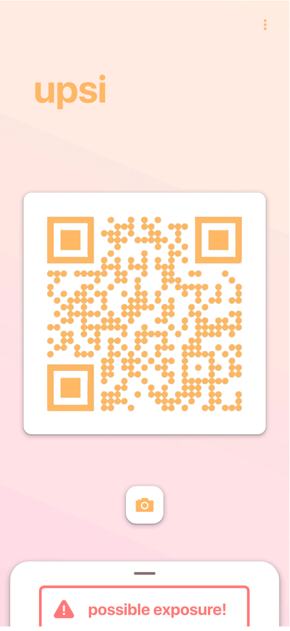
password
****

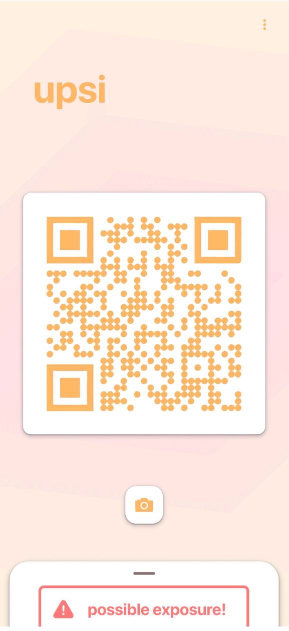


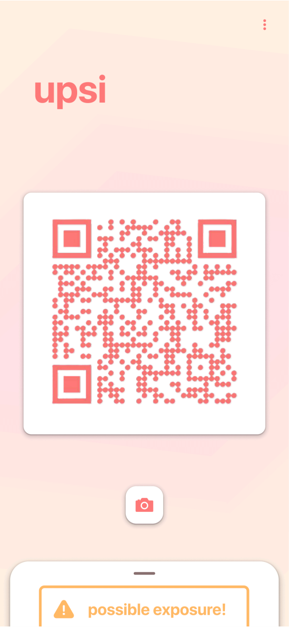


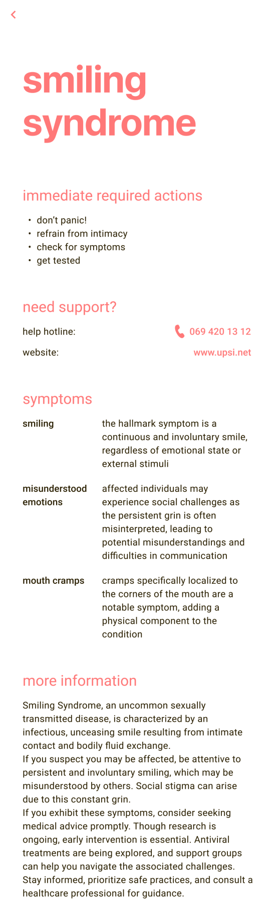












smiling syndrome

immediate required actions

- don’t panic!
- refrain from intimacy
- check for symptoms
- get tested

need support?

help hotline: ☎ 069 420 13 12

website: www.upsi.net

symptoms

smiling the hallmark symptom is a continuous and involuntary smile, regardless of emotional state or external stimuli

misunderstood emotions affected individuals may experience social challenges as the persistent grin is often misinterpreted, leading to potential misunderstandings and difficulties in communication

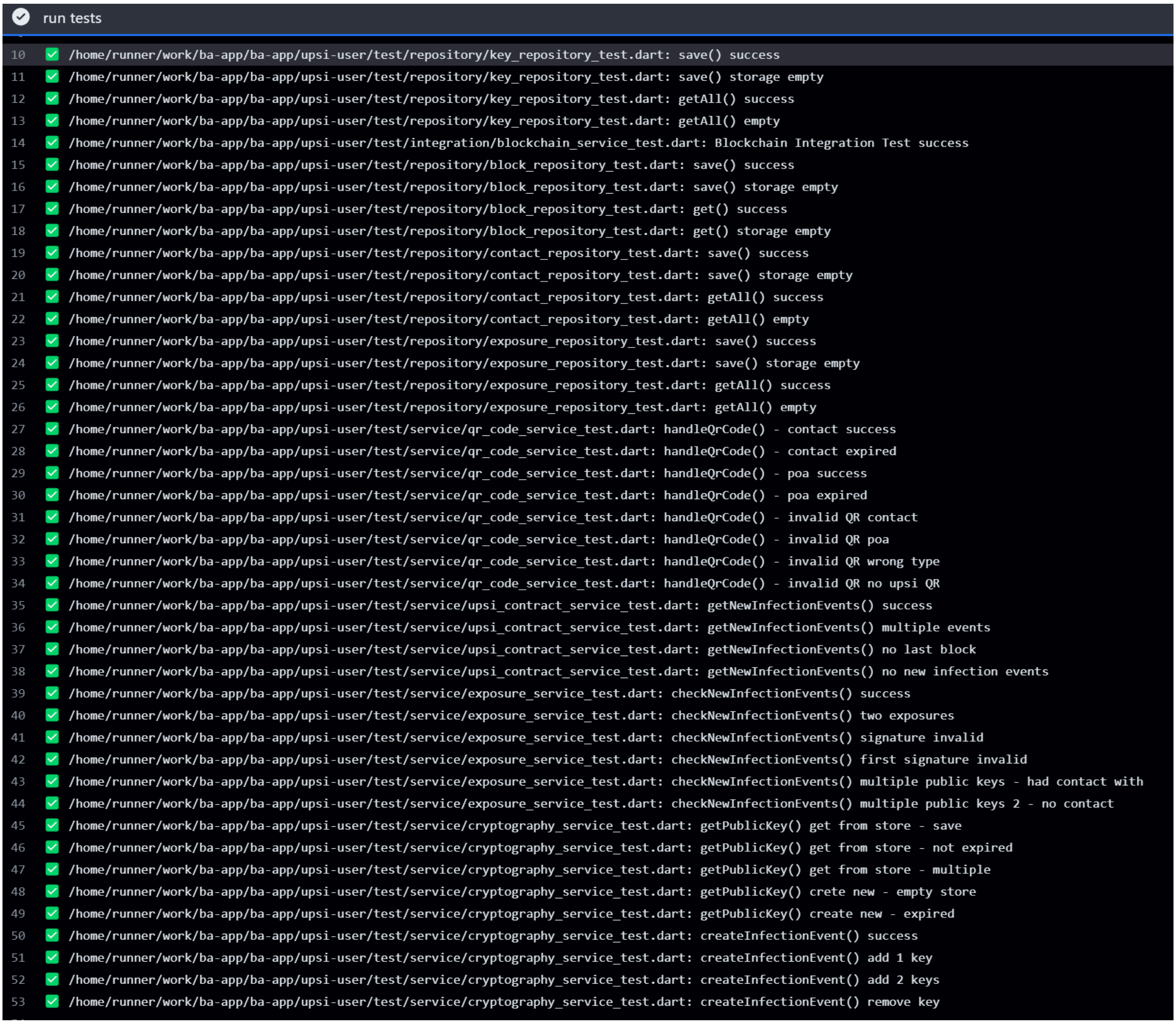
mouth cramps cramps specifically localized to the corners of the mouth are a notable symptom, adding a physical component to the condition

more information

Smiling Syndrome, an uncommon sexually transmitted disease, is characterized by an infectious, unceasing smile resulting from intimate contact and bodily fluid exchange.
If you suspect you may be affected, be attentive to persistent and involuntary smiling, which may be misunderstood by others. Social stigma can arise due to this constant grin.
If you exhibit these symptoms, consider seeking medical advice promptly. Though research is ongoing, early intervention is essential. Antiviral treatments are being explored, and support groups can help you navigate the associated challenges.
Stay informed, prioritize safe practices, and consult a healthcare professional for guidance.

11.6 Unit Tests

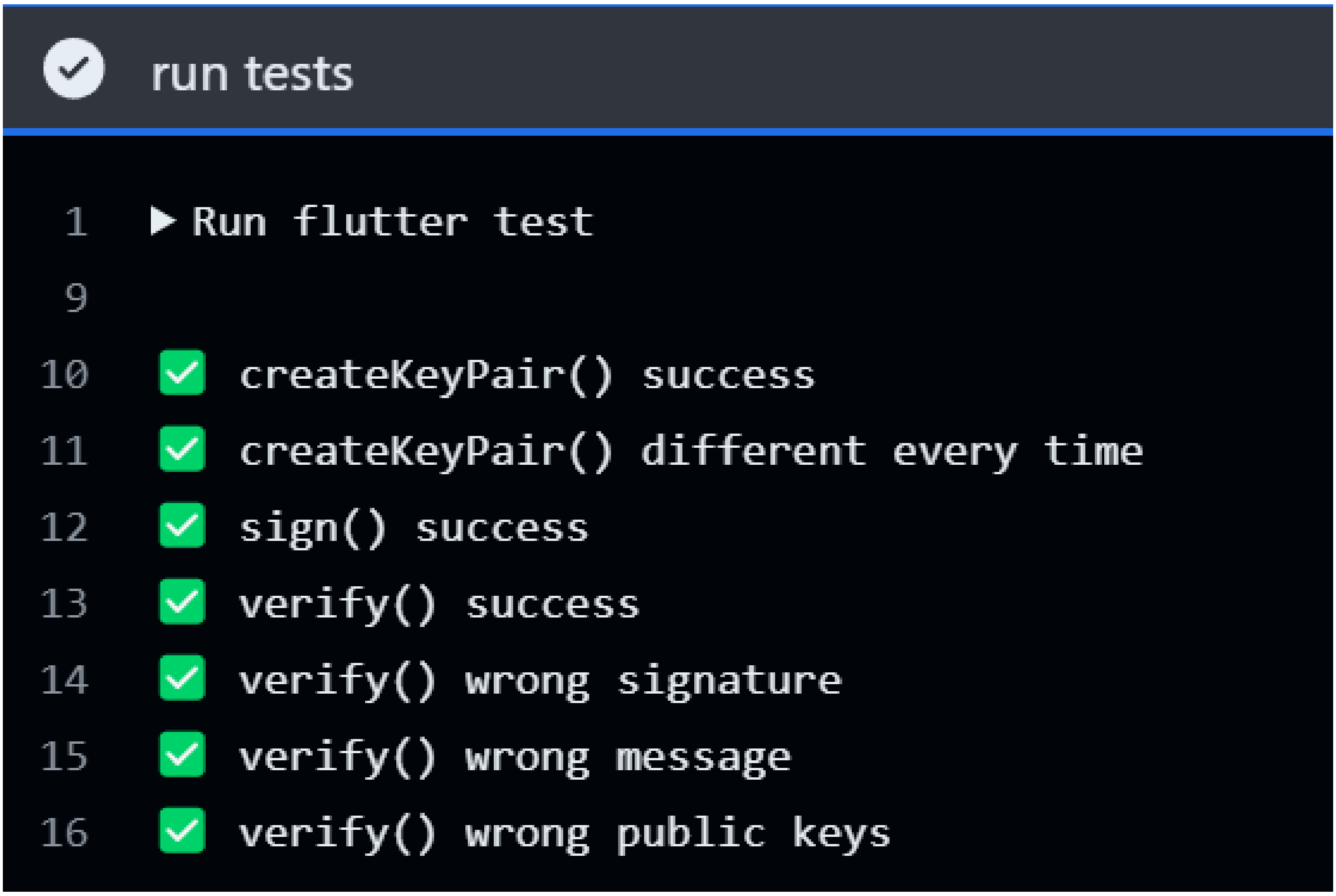
11.6.1 User App



run tests

```
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/repository/key_repository_test.dart: save() success
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/repository/key_repository_test.dart: save() storage empty
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/repository/key_repository_test.dart: getAll() success
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/repository/key_repository_test.dart: getAll() empty
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/integration/blockchain_service_test.dart: Blockchain Integration Test success
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/repository/block_repository_test.dart: save() success
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/repository/block_repository_test.dart: save() storage empty
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/repository/block_repository_test.dart: get() success
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/repository/block_repository_test.dart: get() storage empty
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/repository/contact_repository_test.dart: save() success
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/repository/contact_repository_test.dart: save() storage empty
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/repository/contact_repository_test.dart: getAll() success
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/repository/contact_repository_test.dart: getAll() empty
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/repository/exposure_repository_test.dart: save() success
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/repository/exposure_repository_test.dart: save() storage empty
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/repository/exposure_repository_test.dart: getAll() success
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/repository/exposure_repository_test.dart: getAll() empty
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/qr_code_service_test.dart: handleQrCode() - contact success
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/qr_code_service_test.dart: handleQrCode() - contact expired
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/qr_code_service_test.dart: handleQrCode() - poa success
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/qr_code_service_test.dart: handleQrCode() - poa expired
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/qr_code_service_test.dart: handleQrCode() - invalid QR contact
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/qr_code_service_test.dart: handleQrCode() - invalid QR poa
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/qr_code_service_test.dart: handleQrCode() - invalid QR wrong type
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/qr_code_service_test.dart: handleQrCode() - invalid QR no upsi QR
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/upsi_contract_service_test.dart: getNewInfectionEvents() success
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/upsi_contract_service_test.dart: getNewInfectionEvents() multiple events
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/upsi_contract_service_test.dart: getNewInfectionEvents() no last block
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/upsi_contract_service_test.dart: getNewInfectionEvents() no new infection events
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/exposure_service_test.dart: checkNewInfectionEvents() success
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/exposure_service_test.dart: checkNewInfectionEvents() two exposures
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/exposure_service_test.dart: checkNewInfectionEvents() signature invalid
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/exposure_service_test.dart: checkNewInfectionEvents() first signature invalid
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/exposure_service_test.dart: checkNewInfectionEvents() multiple public keys - had contact with
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/exposure_service_test.dart: checkNewInfectionEvents() multiple public keys 2 - no contact
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/cryptography_service_test.dart: getPublicKey() get from store - save
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/cryptography_service_test.dart: getPublicKey() get from store - not expired
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/cryptography_service_test.dart: getPublicKey() get from store - multiple
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/cryptography_service_test.dart: getPublicKey() crete new - empty store
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/cryptography_service_test.dart: getPublicKey() create new - expired
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/cryptography_service_test.dart: createInfectionEvent() success
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/cryptography_service_test.dart: createInfectionEvent() add 1 key
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/cryptography_service_test.dart: createInfectionEvent() add 2 keys
✅ /home/runner/work/ba-app/ba-app/upsi-user/test/service/cryptography_service_test.dart: createInfectionEvent() remove key
```

11.6.2 Core Package



run tests

```
▶ Run flutter test

✓ createKeyPair() success
✓ createKeyPair() different every time
✓ sign() success
✓ verify() success
✓ verify() wrong signature
✓ verify() wrong message
✓ verify() wrong public keys
```