**Team Development Project: NHS Well-being App**

ICT Project Management in Practice & Law & Technology

Student Names: Irfan Essa, Ella Folan-Bell, Shahil Rahman, Nadifa Hassan, Sabbir Ahamed Saown, Yusuf Muysinkhanov

Student IDs (respectively): 4008609, 3909000, 4011945, 4037051, 4033579, 4031525

Supervised by

Lucia Otoyo & Andy Unger



London South Bank University

Department of Computer Science, Engineering

Team 5

# **i. Acknowledgments**

As collective team members, we would like to express our great gratitude to lectures Lucia Otoyo and Andy Unger for their guidance and supervision on this project. Our clients, Khristina and Tom were also a great help, especially in providing well-being resources as well as keeping constant communication via email. The feedback received by our NHS clients at the end of our sprints allowed us to develop the application to our best standards and as close to the client brief/requirements as possible.

# **ii. List of Figures**

Figure 1: User Roles table .....................................................................................................8

Figure 2: High-Level requirements table........................................….....................................9

Figure 3: Comparing frontend technology stacks..................................................................15

Figure 4: Comparing backend technology stacks..................................................................15

Figure 5: Comparing native technology stacks......................................................................16

Figure 6: Comparing databases.............................................................................................16

Figure 7: Team roles and responsibilities table......................................................................19

Figure 8: Project management tools link................................................................................20

Figure 9: First design - Initial UI ............................................................................................22

Figure 10: Second design - Improved based on client feedback...........................................23

Figure 11: Final design – health and wellbeing pages ..........................................................24

Figure 12: Final design – reporting pages..............................................................................24

Figure 13: Final design – events pages ................................................................................25

Figure 14: Demo video...........................................................................................................26

Figure 15: Regression log testing...........................................................................................27

Figure 16: Product Backlog Snippet One................................................................................29

Figure 17: Product Backlog Snippet Two................................................................................30

Figure 18: Product Backlog Snippet Three.............................................................................31

Figure 19: User Story Snippet.................................................................................................32

Figure 20: Sprint One Snippet.................................................................................................32

Figure 21: Sprint Two Snippet.................................................................................................33

Figure 22: Sprint Three Snippet..............................................................................................33

Figure 23: Sprint Four Snippet.................................................................................................34

Figure 24: Sprint one retrospective meeting review.................................................................34

Figure 25: Sprint two retrospective meeting review.................................................................35

Figure 26: Sprint three retrospective meeting review..............................................................35

Figure 27: Sprint four retrospective meeting review…………………………………..………….36

Figure 28: AI declaration…………………………………………………………..……..………….38

**Table of Contents**

[i. Acknowledgments 2](#_Toc153547148)

[ii. List of Figures 3](#_Toc153547149)

[1. Introduction 7](#_Toc153547150)

[2. Project Overview 7](#_Toc153547151)

[2.1. Aims 7](#_Toc153547152)

[2.2. Background and problem 7](#_Toc153547153)

[2.3. Proposed solution 7](#_Toc153547154)

[2.4. User Roles 8](#_Toc153547155)

[2.5. High-level Requirements 9](#_Toc153547156)

[3. Technical Review 10](#_Toc153547157)

[3.1. Evaluation criteria 10](#_Toc153547158)

[3.2. Research 10](#_Toc153547159)

[3.2.1 Frontend 10](#_Toc153547160)

[3.2.2 Backend 12](#_Toc153547161)

[3.2.3. Native 13](#_Toc153547162)

[3.2.4. Database 14](#_Toc153547163)

[3.3. Evaluation 15](#_Toc153547164)

[3.3.1. Frontend Comparison 15](#_Toc153547165)

[3.3.2. Backend Comparison 15](#_Toc153547166)

[3.3.3. Native Comparison 16](#_Toc153547167)

[3.3.4. Database Comparison 16](#_Toc153547168)

[3.4. Final Decision and Justification 17](#_Toc153547169)

[4. Team Structure, roles and responsibilities 17](#_Toc153547170)

[4.1. Team Structure 17](#_Toc153547171)

[4.2. Roles and Responsibilities: 18](#_Toc153547172)

[4.3. Project Management Tools 20](#_Toc153547173)

[5. Design 20](#_Toc153547174)

[5.1. Requirements 20](#_Toc153547175)

[5.2. First Design 22](#_Toc153547176)

[5.2. Second Design 23](#_Toc153547177)

[5.3. Final Design 24](#_Toc153547178)

[6. Development 26](#_Toc153547179)

[7. Product and Sprint backlogs 29](#_Toc153547180)

[7.1. Product Backlog 29](#_Toc153547181)

[7.2. User Story 32](#_Toc153547182)

[7.3. Sprints 32](#_Toc153547183)

[7.3.1. Sprint 1 32](#_Toc153547184)

[7.3.2. Sprint 2 33](#_Toc153547185)

[7.3.3. Sprint 3 33](#_Toc153547186)

[7.3.4. Sprint 4 34](#_Toc153547187)

[8. Retrospective meetings 34](#_Toc153547188)

[8.1. Meeting – Sprint 1 34](#_Toc153547189)

[8.2. Meeting – Sprint 2 35](#_Toc153547190)

[8.3. Meeting – Sprint 3 35](#_Toc153547191)

[8.4. Meeting – Sprint 4 36](#_Toc153547192)

[10. References 37](#_Toc153547193)

[11. AI Statement 38](#_Toc153547194)

# **1. Introduction**

The following report outlines the research, design, implementation, and testing of an NHS well-being app proposed by St George’s Hospital. The system enables NHS staff to report bullying and harassment complaints. This, in turn, enables the staff at St George’s Trust to improve their mental health, and as a result, provide better service to patients. It will also likely reduce staff turnover and increase employee recommendations/referees (resulting in a decrease in recruitment costs to the NHS).

# **2. Project Overview**

## **2.1. Aims**

This project aims to design and implement a mobile application that allows the user to access available resources and will allow the staff to easily access support.

## **2.2. Background and problem**

Since 2016, the St George’s University Hospitals NHS Foundation Trust has been operating as a ‘trust that has 9.5k employees spread around multiple sites. The employees that work within the trust are very diverse in their background and the jobs that they do” (NHS Foundation Trust, *About Us - St Georges*). The trust, like many others, has been educating its staff on their practices surrounding bullying and harassment in the workplace.

The system is designed to allow the employees to have a space where issues of bullying and harassment that occur in the workplace are anonymously reported. The system in place will also aim to promote the staff’s wellbeing, this is done by providing the users with a health and wellbeing page where these resources will be available. On the application, there is also an event page that will aid in staff training and wellbeing.

## **2.3. Proposed solution**

From the brief that was given to us by the client, they asked that when coming up with a solution we seek inspiration from the “UCL Report + Support” page. Their page provides links and videos to provide “more information and guidance about making a report and receiving support”. Having links and videos means that finding information on how to report wellbeing is far less time-consuming which is an objective that we work through in development, this is important as NHS staff have short breaks and will need to be able to access information in a fast manner.

The mobile application is intended to be a tool that is used within the trust and will contain the relevant resources to support the employees. The solution is required to authenticate the users that access the mobile application, to do this we have used the NHS trust email that is provided to all staff members. Once the user has logged in the system will guide the user through the features that the mobile application has through a user-friendly manner: dropdowns, links, clear headings, search functions and predictive text. By prioritising a user-friendly application, it means that the system can be accessible to all users. The system will interactively provide training for the staff, providing information on the guidelines for bullying and harassment in the workplace. When reporting a claim, the application will take the user through the guidelines of what bullying, and harassment is in the workplace, allowing them to further their claim if the correct criteria are met.

## **2.4. User Roles**

A screenshot of a computer

Description automatically generated Figure 1: User roles table

## **2.5. High-level Requirements**

A screenshot of a white and black page

Description automatically generated Figure 2: High-Level requirements table

# **3. Technical Review**

This chapter will outline the research conducted on potential technology stacks along with   
their pros and cons, based on the evaluation criteria specified in section 3.1. The justification section will outline a high-level evaluation from which we based our final decision.

## **3.1. Evaluation criteria**

The following were considered when considering potential technology stacks (Ranked in order of priority):

1. Application size – The software must not be larger than 100MB, considering NHS Wi-Fi speeds may be slow
2. Performance – The final software/product should not lag (drop below 20fps) on a minimum 1.2GHz CPU
3. Documentation and community support – There must be significant public resources available to aid development and maintenance
4. Team expertise – The chosen stack should be within the team's skill scope and areas of interest.
5. Cross-platform - The application should be cross-platform since NHS staff run both IOS and Android
6. Cost – The maintenance cost should be below £200 per month (I.e. cost of labour for future maintenance)

## **3.2. Research**

### **3.2.1 Frontend**

#### **3.2.1.1 Research into Flutter**

(Bernat, 2023) states in his article that Flutter can be used to create cross-platform (I.e. android/iOS/PC/mac) applications using one code base. This would significantly reduce development time, likely leading to a decrease in cost for our NHS clients. He also develops in his analysis that Flutter has overtaken react native in terms of popularity in 2021. This rise in popularity would allow us, as developers, to have great access to documentation and community forms which would in turn aid in implementation and debugging.

**Advantages of flutter**

One benefit of using Flutter is its "hot reload" function which stands out for its ability to allow developers to instantly preview changes without restarting the app, which significantly accelerates the development workflow (Bachelder, 2020).

Another edge that Flutter has over alternatives is its use of Google’s Dart programming language which compiles directly to machine code (Wightwick, 2021), hence significantly improving performance.

One benefit of using Flutter is its "hot reload" function which stands out for its ability to allow developers to instantly preview changes without restarting the app, which significantly accelerates the development workflow (Bachelder, 2020).

**Disadvantages of flutter**

A commonly known disadvantage of flutter is its tendency to produce larger-sized apps. This would increase download times. It is also worth noting that if the application is loaded on a webpage/browser, a higher internet speed may also be required.

#### **3.2.1.2. Research into react native**

React native is a JavaScript-based application development tool created by Facebook. The singular piece of code written can be used on both Android and iOS platforms. (Paterska, 2023) also explains that react native utilises a bridge concept to allow for asynchronous communication between JavaScript and native elements.

**Advantages of React native**

A key advantage of react native is its ability to effectively reuse code.

**Disadvantages of React native**

A disadvantage react native carries over Flutter is its comparative lower ranking when sorted by popularity. This is shown in Stack Overflow’s yearly survey (Stack Overflow, n.d.) where react native was used by 8.43% of developers, whereas flutter was used by 9.12%.

### **3.2.2 Backend**

#### **3.2.2.1. Research into Node JS / Express**

Express is a NodeJS framework used to create mobile and web applications. It can be used for all, single page, multiple page and hybrid web applications (Holmes, 2018). Express is built on top of Node JS which helps manage servers and routing.

**Advantages of Node JS / Express**

One advantage of expressJS is that it is highly scalable. This is due to its event-driven and non-blocking I/O model, which in turn allows it to handle multiple requests at the same time.

Another important advantage is its use of JS, which is relatively easy to learn, especially considering the team's background in React. This would reduce risk and unpredictability when meeting key development deadlines.

**Disadvantages of Node JS / Express**

(Sharma, 2023) points out that there are usually many issues with callbacks involved with Express. This is also worsened by the fact that error messages quickly start to become difficult to understand.

#### **3.2.2.2. Research into Python/Django**

Django is Python's framework which is typically used for back-end web development. (Holovaty & Kaplan-Moss, 2021) further explains Django's use of the Don’t repeat yourself (DRY) principle which promotes code reusability and reduces redundancy.

**Advantages of Python/Django**

A key advantage of utilising Django for the backend, especially in this NHS project, is its high community support. This is particularly important as we would like to reduce maintenance costs for our NHS clients due to its lack of funding.

**Disadvantages of Python/Django**

The main disadvantage of Django for this well-being project is the fact that it requires a longer time to initially set up the framework environment. This would be a large hindrance because many developers are working on this project on their independent computers. Another reason this is a noteworthy disadvantage is because of the tight time constraints of this project.

### **3.2.3. Native**

#### **3.2.2.1. Research into Swift**

Swift is a general-purpose programming language designed by Apple and aimed to be easy to learn, safe (achieved through the elimination of classes and unsafe code) and powerful. (Swift, 2023) even states that it’s on par with C in terms of performance. It also provides inferred types which improves on code readability and minimises mistakes.

**Advantages of swift**

As stated above, swift’s main advantage is its safety, this is achieved for example through the use of optional types which aid in preventing null pointer exceptions. Another key advantage of Swift is its performance. (Johnson, 2020) states this is due to its optimised compiler which tends to be more efficient than objective C.

**Disadvantages of swift**

A vital disadvantage of Swift is its lack of support for Android/Linux-based applications. This would be a significant hurdle as St Geroge's NHS staff is very diverse and is likely to have a split of mobile operating systems.

#### **3.2.2.2. Research into Kotlin**

Kotlin is a multiplatform language which is particularly good for Android and Java development (JetBrains, 2021). It also utilises simple syntax and includes safety precautions resulting in a reduction in bugs.

**Advantages of Kotlin**

The main advantage of Kotlin, as explained above, is its null safety. This reduces the likelihood of null pointer exceptions which are common in many languages such as Java.

**Disadvantages of Kotlin**

The main advantage of Kotlin, similar to Swift, is its limited cross-platform capabilities when compared to languages such as Python/Django. Kotlin is usually used for Android and JVM-based environments.

### **3.2.4. Database**

#### **3.2.4.1 Research into MongoDB**

MongoDB is known for its scalability and flexibility. It also uses a document-oriented approach for data storage, making it highly suitable for dynamic and large-scale applications (MongoDB, Inc., 2023).

**Advantages of MongoDB**

An advantage of MongoDB that would be beneficial to us is its dynamic schema design enables rapid iteration and development, making it ideal for applications that require quick adaptation to changing data requirements.

**Disadvantages of MongoDB**

A key disadvantage as stated by (Banker, 2021) is MongoDB’s slow performance. This is due to its document-oriented model, as well as the fact that it utilises an aggregation framework.

#### **3.2.4.2 Research into Postgres SQL**

Postgres SQL is an open-source, object-relational database. It supports both SQL (relational) and JSON (Non-relational queries). Postgres is also acid-compliant meaning it follows principles of Atomicity, Consistency, Isolation, and durability, which all aid in ensuring data integrity.

**Advantages of Postgres SQL**

A key advantage is Postgres's Concurrency Control (MVCC) feature which allows for concurrent processing of multiple transactions without locking the database. This in turn significantly improves performance and scalability (Obe & Hsu, 2021).

A further advantage of its use of ACID is shown and explained in section 3.2.3.

**Disadvantages of Postgres SQL**

A disadvantage of Postgres is due to its use of ACID. It causes high overhead and struggles in terms of performance, particularly when dealing with a high volume of reads and writes. However, it is worth noting that when compared to MongoDB, Postgres’s performance, when dealing with a smaller frequency of read/writes (as will be in this project) is noticeably better.

## **3.3. Evaluation**

To conduct a more objective and measurable evaluation which would aid in our decision-making and justification on which technology stacks to pick, we have implemented a numbering/marking system for each stack against the criteria shown in section 3.1

3 = Strong

2 = adequate

1 = Weak

### **3.3.1. Frontend Comparison**

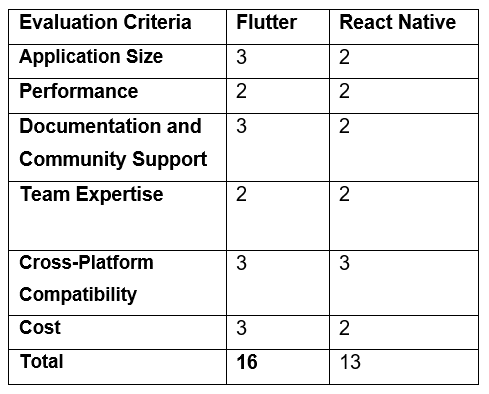


Figure 3: Comparing frontend technology stacks

### **3.3.2. Backend Comparison**

A table of code with black text

Description automatically generated

Figure 4: Comparing backend technology stacks

### **3.3.3. Native Comparison**

A white sheet with black text

Description automatically generated

Figure 5: Comparing Native technology stacks

### **3.3.4. Database Comparison**

A white sheet with black text

Description automatically generated

Figure 6: Comparing databases

## **3.4. Final Decision and Justification**

Based on team collaboration in determining the scores above guided by the research conducted in section 3.2, we have decided on Flutter for the frontend, Note.js/express for the backend and PostgreSQL for the database. Upon researching native technology stacks, they initially seemed attractive due to their high performance, and the ability to minimise bugs, however upon further discussion, we decided to disregard them as a potential solution due to their lack of support for cross-platform applications. As stated previously, this was a key criteria especially when considering St Georges's diverse staff members. We then discussed our choices with our NHS clients in person where they agreed that these technology stacks would be most appropriate due to their high performance, cross-platform compatibility and vast established documentation & community support.

# **4. Team Structure, roles and responsibilities**

## **4.1. Team Structure**

In our project, Team 5 adopted the Agile Scrum methodology, which facilitated a non-hierarchical team structure. Instead of traditional roles and responsibilities, we embraced a distributed approach that leveraged the individual strengths and weaknesses of each team member.

Following the Scrum framework, Linda, our client, assumed the crucial role of the Product Owner. As the Product Owner, Linda was responsible for defining and prioritising project requirements, ensuring alignment with stakeholder needs, and maximising the value delivered by the team.

Andy Unger, our tutor, took on the role of the Scrum Master. As the Scrum Master, Andy served as the facilitator and guardian of the Scrum process. Their primary responsibilities included guiding the team in adhering to Scrum principles, removing any impediments that hindered progress, and fostering a collaborative and self-organizing environment with his direct feedback.

Team 5, comprising of dedicated and cross-functional professionals, constituted the Scrum Development Team. This team possesses the necessary skills and expertise to deliver the project increment. They work collectively, leveraging effective communication and close collaboration to ensure the successful completion of the project.

By adopting the Scrum methodology, our team embraced the core Scrum events, including the Daily Stand-up, Sprint Planning, Sprint Review, and Sprint Retrospective. These events facilitated regular communication, efficient planning, progress review, and continuous improvement within our team.

The Agile Scrum methodology, with its emphasis on adaptability, transparency, and collaboration, enabled our team to effectively manage the project. By leveraging the strengths of each team member and adhering to Scrum principles, we were able to deliver a high-quality product that met the expectations of our client and stakeholders.

## **4.2. Roles and Responsibilities:**

To allocate roles and responsibilities within our team, we followed a systematic two-step process. Initially, we assessed our Belbin Roles to identify our individual strengths and areas of expertise. This allowed us to gain insights into our team's collective skill set. Subsequently, we created a Confluence page as a collaborative platform where each team member documented their strengths and expressed their desire to improve in specific roles throughout the project.

By engaging in open discussions regarding our strengths and aspirations, we collectively assigned roles that aligned with our individual preferences and capabilities. To ensure a smooth development process, we decided to commence development from the second sprint. Consequently, we divided the development tasks into two groups. Yusuf and Nadifa assumed responsibility for backend development, leveraging their expertise in that area. Meanwhile, Sabbir, Ella and Irfan took charge of front-end app development tasks, capitalising on their skills in that domain. Given Nadifa's demonstrated design proficiency, she also led the interface design efforts for the application.

To maintain a high standard of quality, we implemented a review board system. Each task was assigned to a team member, who was then responsible for reviewing the work of another team member. This process allowed for comprehensive feedback and ensured that tasks were thoroughly evaluated before implementation.

In addition to our development efforts, we recognised the importance of documenting our project in a comprehensive report. Irfan and Ella, who both exhibited strong writing skills and confidence in report writing, were entrusted with the responsibility of documenting the entire project while Shahil was researching the legal side of the project and providing advice as reports. However, it's worth noting that other team members actively contributed to the report and maintained constant communication to prevent any delays in its completion.

By following this structured approach to roles and responsibilities, we aimed to leverage our team's strengths, foster collaboration, and ensure the successful execution of our project.

A white sheet with black text

Description automatically generated

Figure 7: Team roles and responsibilities table

## **4.3. Project Management Tools**

During the length of the project, as a team, we decided to use Jira as our task management and team progress monitoring tool, and we have used Confluence for retrospective documentation. Table 3 provides links to the tools:

|  |  |
| --- | --- |
| Tool | Link |
| Jira | <https://irfanessa.atlassian.net/jira/software/projects/NWAT5/boards/2> |
| Confluence | <https://irfanessa.atlassian.net/wiki/spaces/NWAT5/pages/983041/Team+roles+and+responsibilities> |

Figure 8: Project management tools link

# **5. Design**

## **5.1. Requirements**

During the first meeting with the client, we gathered the requirements for the application. They are outlined below:

**Client requirements**

* End users: St. Georges Trust staff members
* Priority on engagement
* Web and app-based tool - Compatible with iOS and Android
* Separate from professional settings
* Easy access for reporting and support
* Client prefers UCL layout
* Simplicity is crucial, more details available via hyperlink
* Feedback system for missing information
* Interactive staff training with linked resources
* Consider diverse users: second language, screen readers, accessibility for all
* Wellbeing resources needed with distinct headings
* Avoid app dropdown menu; display main headings on separate pages
* Include search and predictive text functions for user convenience

**Pages**

* Home page
* Health and wellbeing page
* Events page
* Report bullying and harassment page
  + Report with contact details
  + Report anonymously

**Security**

* Ensuring GDPR compliance and secure data handling is crucial in storing bullying and harassment information. Addressing security design issues is vital for commitment to data protection.
* Guiding users through established protocols is essential when handling reports of bullying and harassment. Providing staff with extra guidelines improves their understanding of the reporting process.
* Balancing anonymity and verification are key. Enabling anonymous reporting while confirming the reporter's hospital affiliation using existing credentials, like NHS login and email, verifies identity without compromising anonymity.
* Implementing a dual reporting process—one for named individuals with claim follow-up and another for anonymous reporting—ensures flexibility.

## **5.2. First Design**

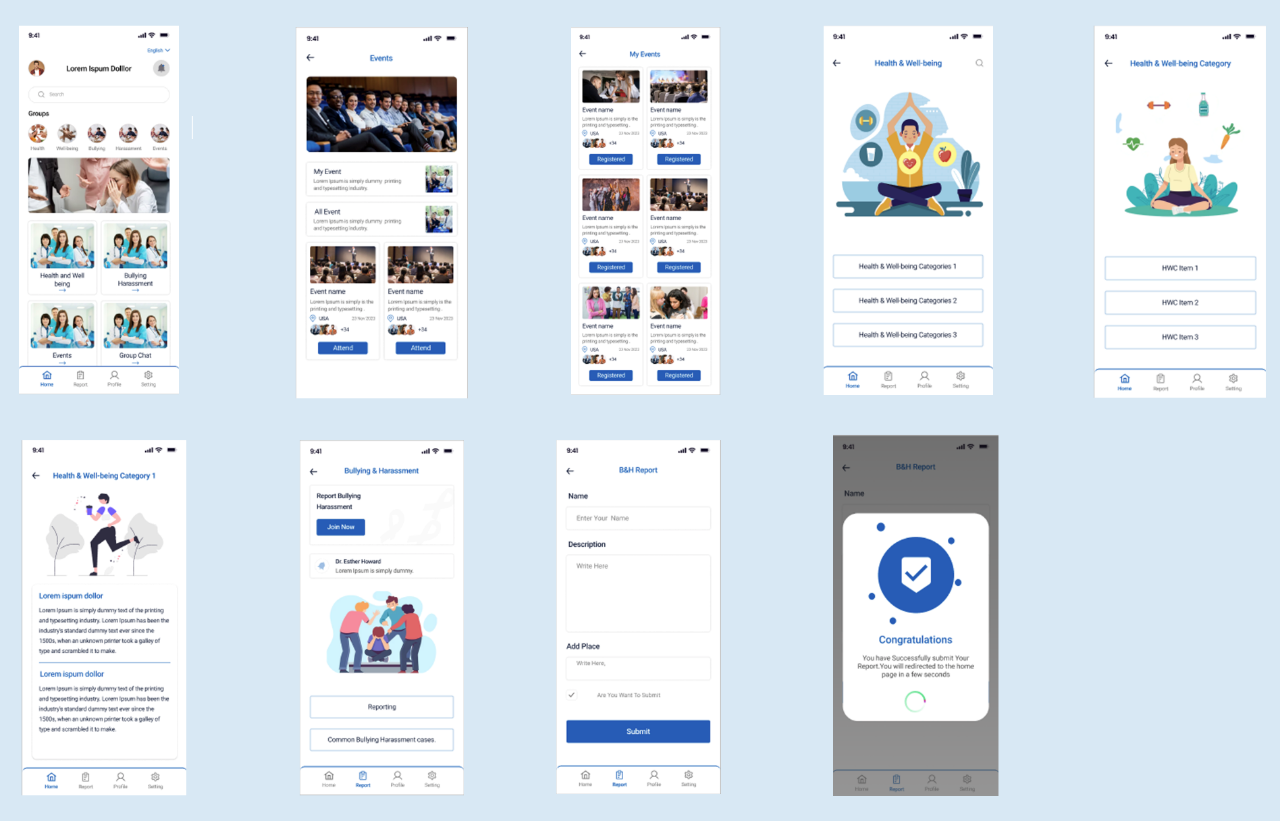
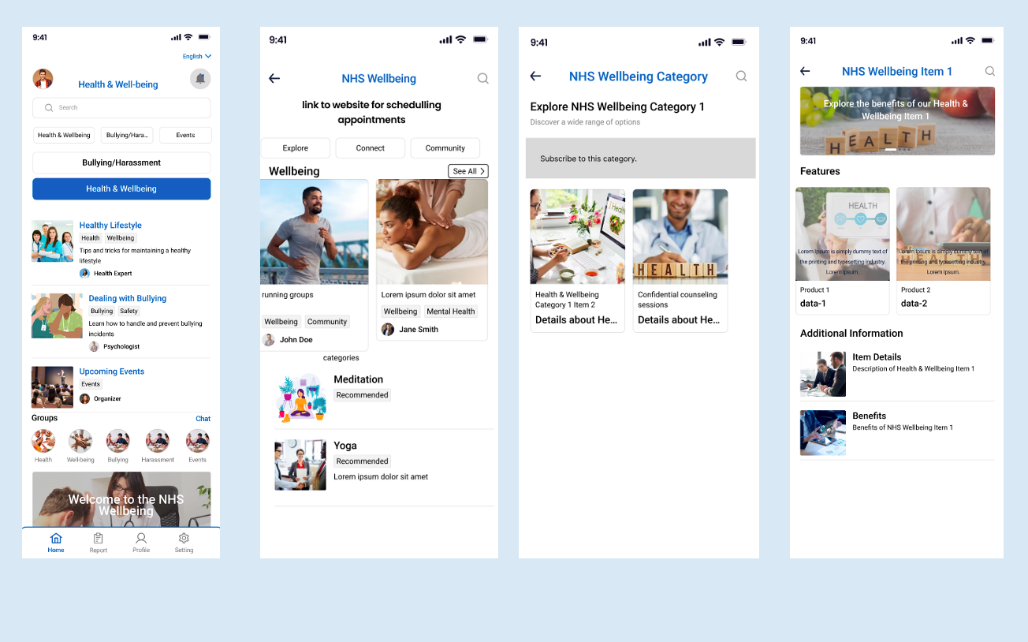


Figure 9: First design - Initial UI

Following the second client meeting we took the requirements for the application that the client wanted, and we created designs based on those, we then received feedback on the presented designs, as shown in the figure below. For the login page, the client expressed a preference for users to log in using their NHS email instead of connecting through Google or Facebook. Regarding the homepage, the client emphasised prioritising the accessibility of well-being resources in the application. As a result, we agreed within the meeting to customise and personalise the notifications sent to users accordingly.

The client's feedback on the bullying and harassment page highlighted the need for clearer distinctions among the reporting buttons. The client also wanted to have clear definitions of bullying and harassment shown to the user, so that they can make an informed decision. Additionally, a concern was raised about how the Human Resources (HR) department would handle the volume of reporting within the application.

## **5.2. Second Design**



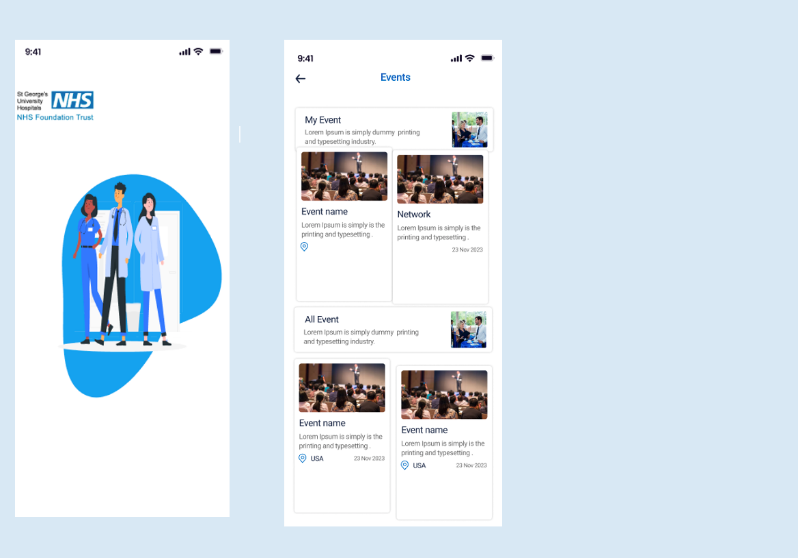
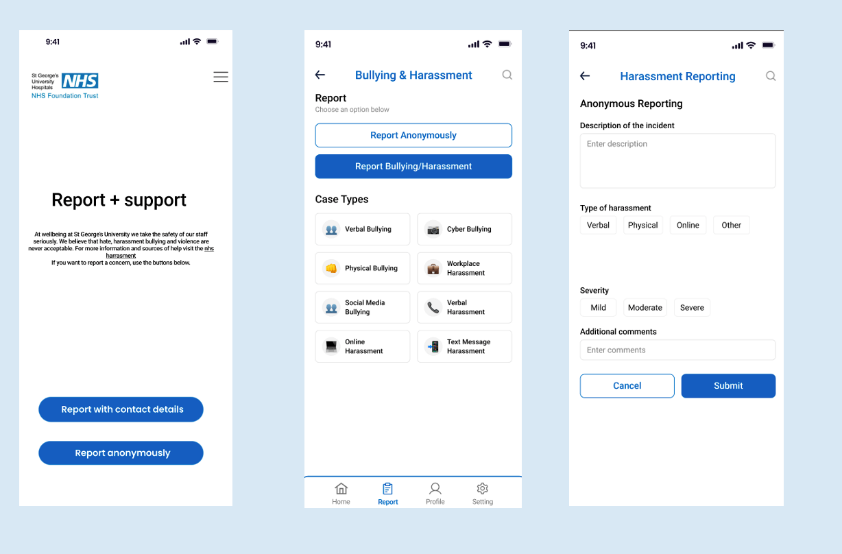


Figure 10: Second design - Improved based on client feedback

Following our third client meeting, we incorporated feedback gathered from the previous sessions to refine our designs, which can be seen in the figures below. The updated designs were presented, and we received feedback indicating a desire for a more interactive approach to health and wellbeing information, including elements like cartoons and videos. The feedback that we received for the reporting page was that it was important for the users to be able to report anonymously or to be named, an aesthetic point on the design that they made was that they wanted the boxes for reporting to be different colours to avoid them looking similar. Additionally, it was noted that the application should be tailored to the specific needs of St. George's Trust staff. Regarding user authentication, a proposal was made for the trust to generate a unique single-use code for users to access the application.

## **5.3. Final Design**

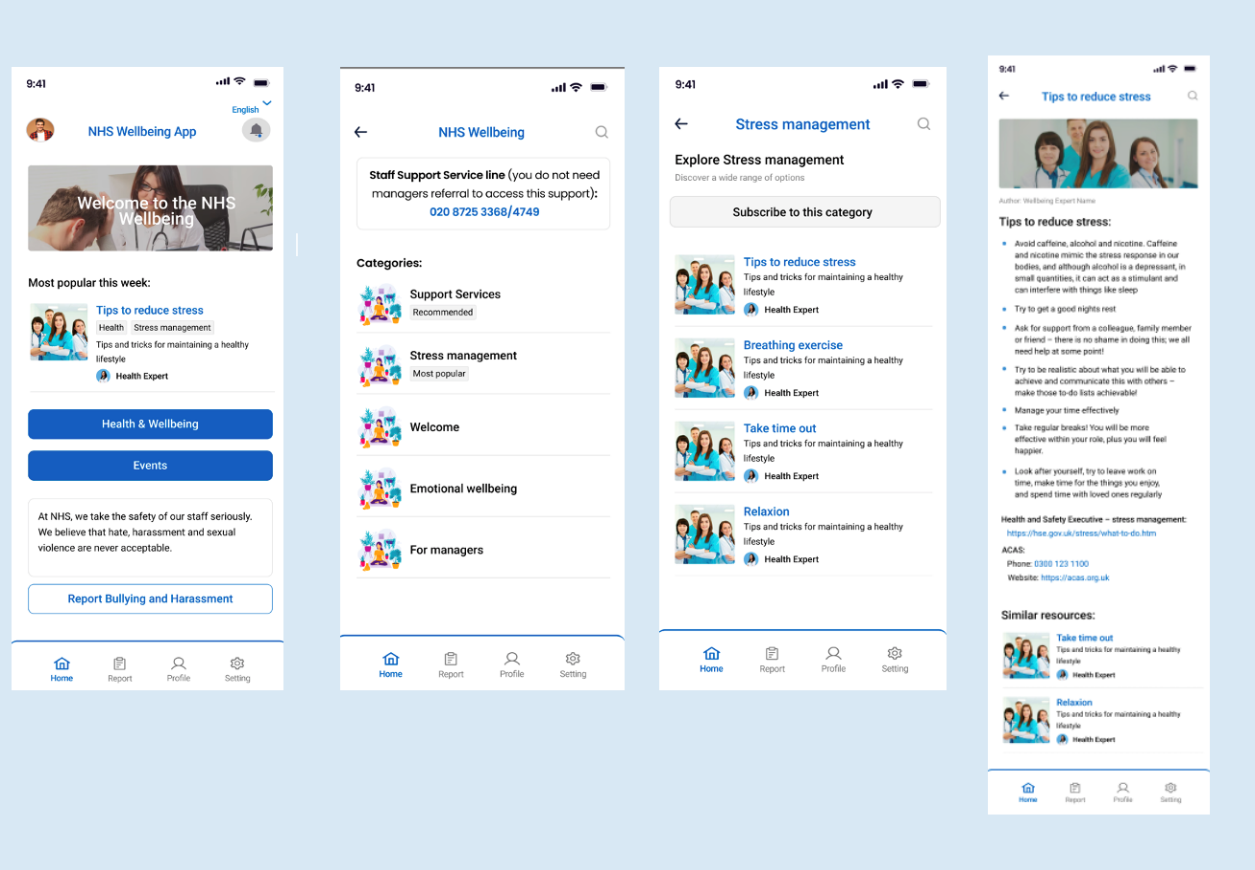


Figure 11: Final design – health and wellbeing pages

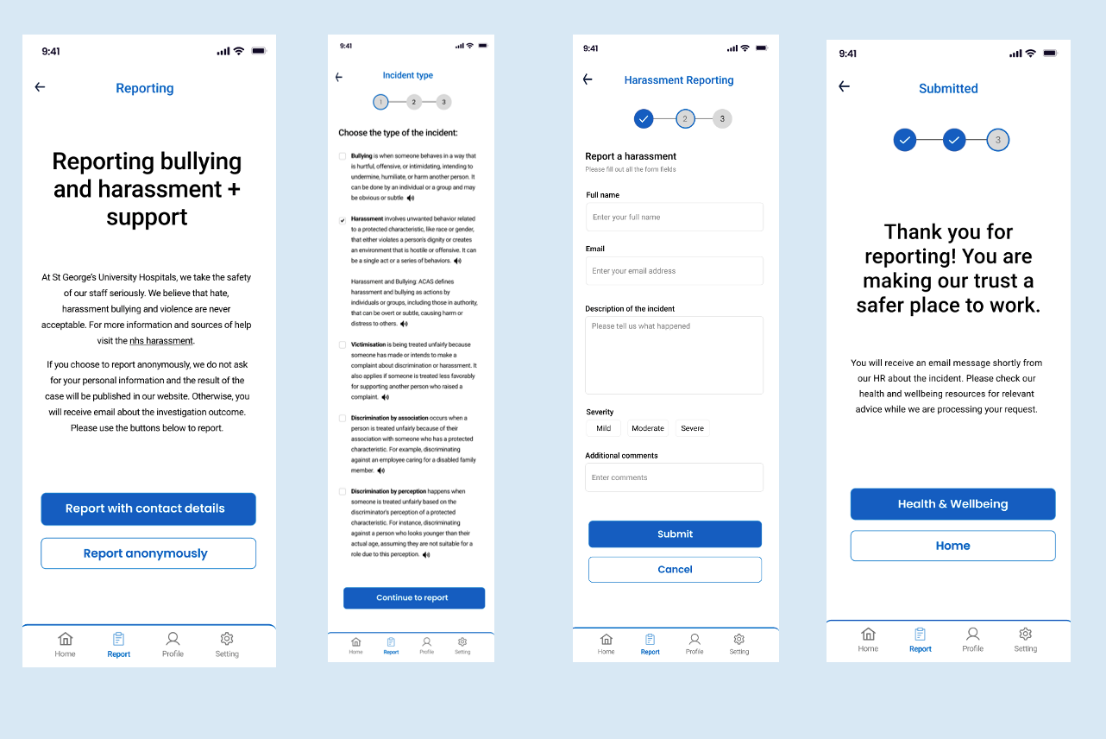


Figure 12: Final design – reporting pages

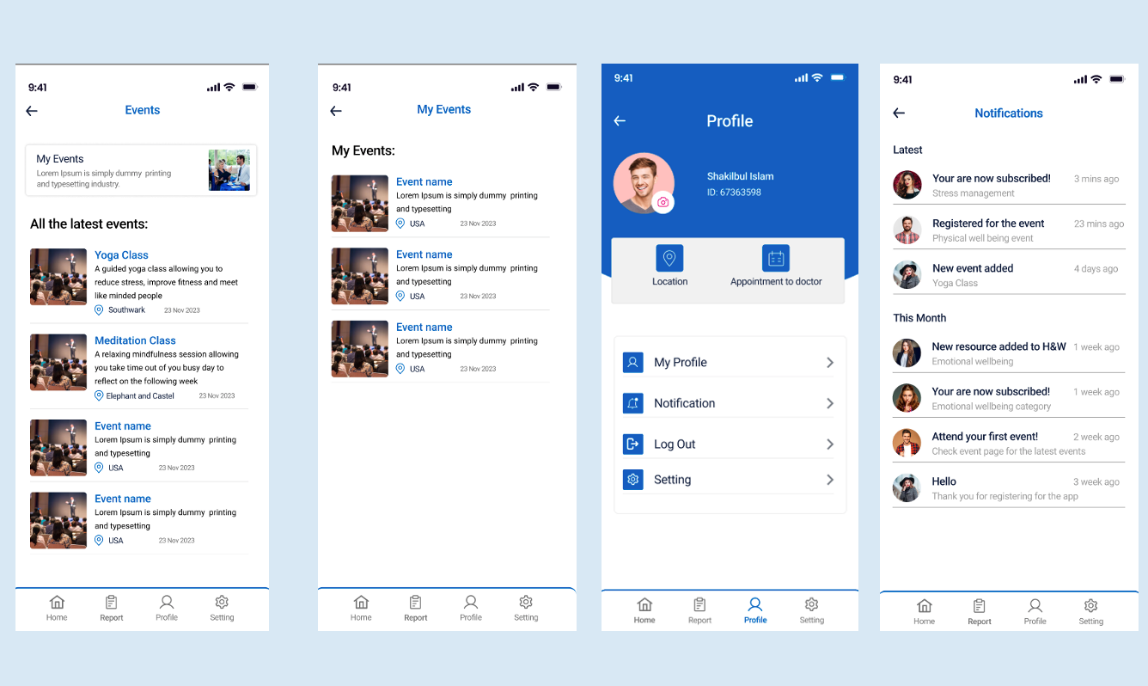


Figure 13: Final design – events pages

In our most recent client meeting, we received conclusive feedback on the design and application we developed, which can be seen in the figures below. The client expressed satisfaction with the use of information derived from the booklet they supplied for the application. Changes that they wanted to be made were that the pictures that we use within the application need to reflect the diversity of the staff members of the trust.

Approval was granted for the reporting page, and the client is pleased with the provided definitions, emphasising that it enables users to make informed choices. A point that was made for the developers of the application was to think about the user perspective regarding subscribing to events, does the subscription only allow them to subscribe to one class or ten, this is a point that will be taken in for further development of the application.

# **6. Development**

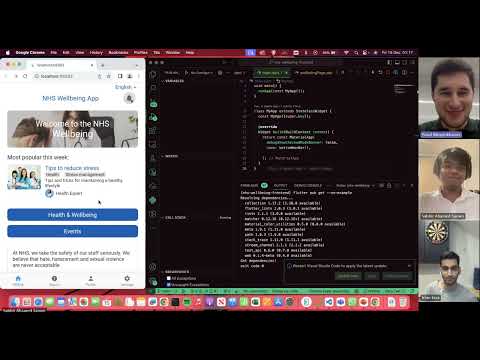
[](https://www.youtube.com/watch?v=D2cmczkmxzs)

Figure 14: Demo video

The link to the video demonstration: <https://www.youtube.com/watch?v=D2cmczkmxzs>

The link to this regression log file: [Regression test log.xlsx](https://stulsbuac.sharepoint.com/:x:/s/Team524/EZXVG1NZ0FxPnUZtHH4ZnAABGCGnxDO1fQOJVclVtjciiQ?e=VL8Al7)

The GitHub repository link: <https://github.com/irfanessa1/nhs-wellbeing-frontend>

This video shows our team presentation for the app and its features. Both the design prototype and the flutter application are demonstrated in the video.

Because of the short time left for the development, the team had to prioritise the most important pages and we chose Homepage, Health and Wellbeing, and Report pages for the implementation. During the app development, the client feedback was invaluable, and we managed to develop the features our clients asked for. Particularly, the clients suggested we implement a search functionality, a section for quick support line access, and a way to subscribe to a category to be notified when new articles are added to well-being resources. The clients also wanted us to separate the report button in homes and add a section mentioning the Trust takes bullying and harassment seriously. They asked us to make the reporting a step-by-step process giving relevant information at each stage.

For these reasons, we made the 4 pages for reporting an incident. On the first page, the app shows information that Trust takes these incidents seriously and buttons for reporting with contact details and anonymous reports which we could not implement due to time constraints. On the second page of reporting, the app gives users a list of incident types and users can select one to proceed to the third page where users fill out the incident form and the app checks for any empty fields or invalid email format and shows error messages in case it finds any. Finally, the last page informs users their request has been submitted, and HR will contact them shortly. In the meantime, users can go back to the home page and explore the health and wellbeing resources.

We made every effort to address client feedback during the development and each one of the dev team had been informed about client meeting notes in the Jira and confluence page for the meeting. The development work was divided between three people:

1. Nadifa – the app homepage

2. Sabbir – the report pages

3. Yusuf – the health and wellbeing pages

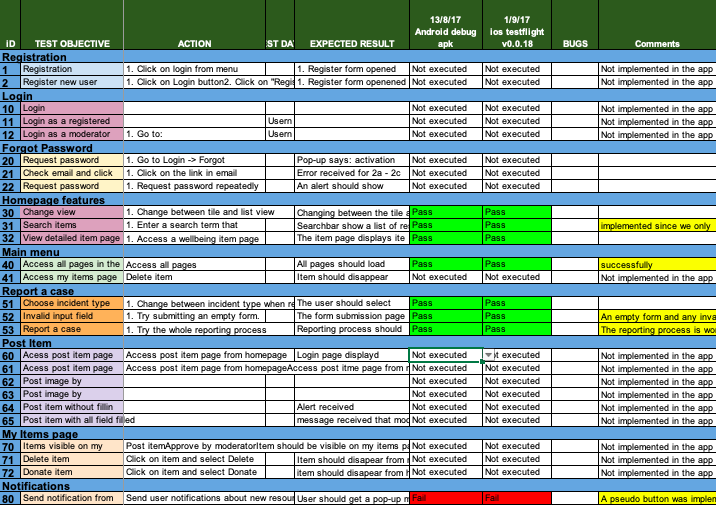


Figure 15: Regression log testing

In the testing of the application, some of the pages from the regression log were not tested since we did not implement those pages due to a short time frame and client feedback. All the pages and features we implemented passed the tests except for sending notifications from devices. There is a button to subscribe to a category on the wellbeing category page, but it does not fully work, and this was the only feature that did not pass the test.

The development team collaborated in coding this Flutter application using Git and GitHub. Initially, we planned to implement both the front-end (flutter) and a back-end server with a database, but we decided to prioritise the user interface. The codebase can be found in the following repository:

# **7. Product and Sprint backlogs**

## **7.1. Product Backlog**



Figure 16: Product Backlog Snippet One



Figure 17: Product Backlog Snippet Two

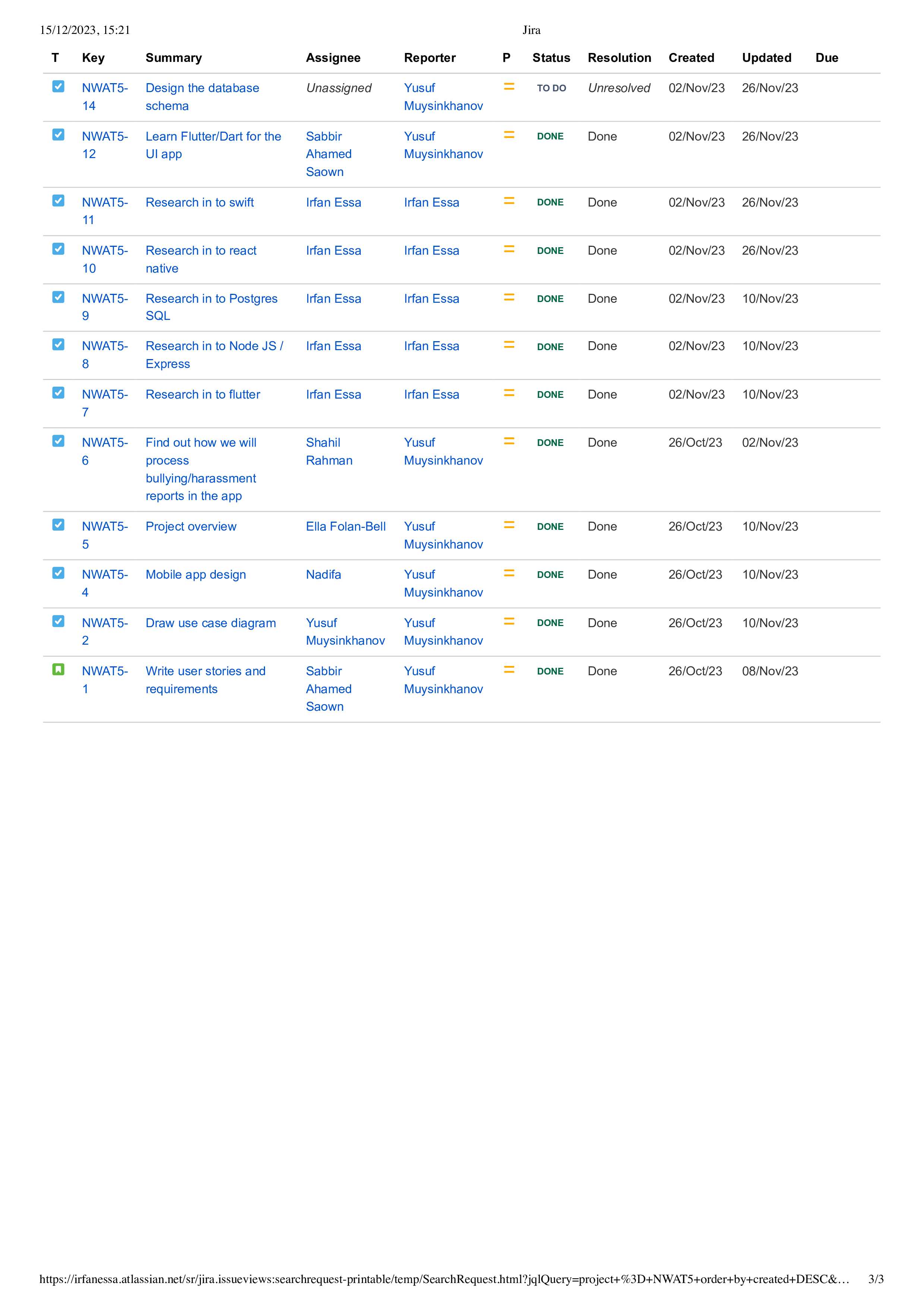


Figure 18: Product Backlog Snippet Three

## **7.2. User Story**

A screenshot of a computer

Description automatically generated

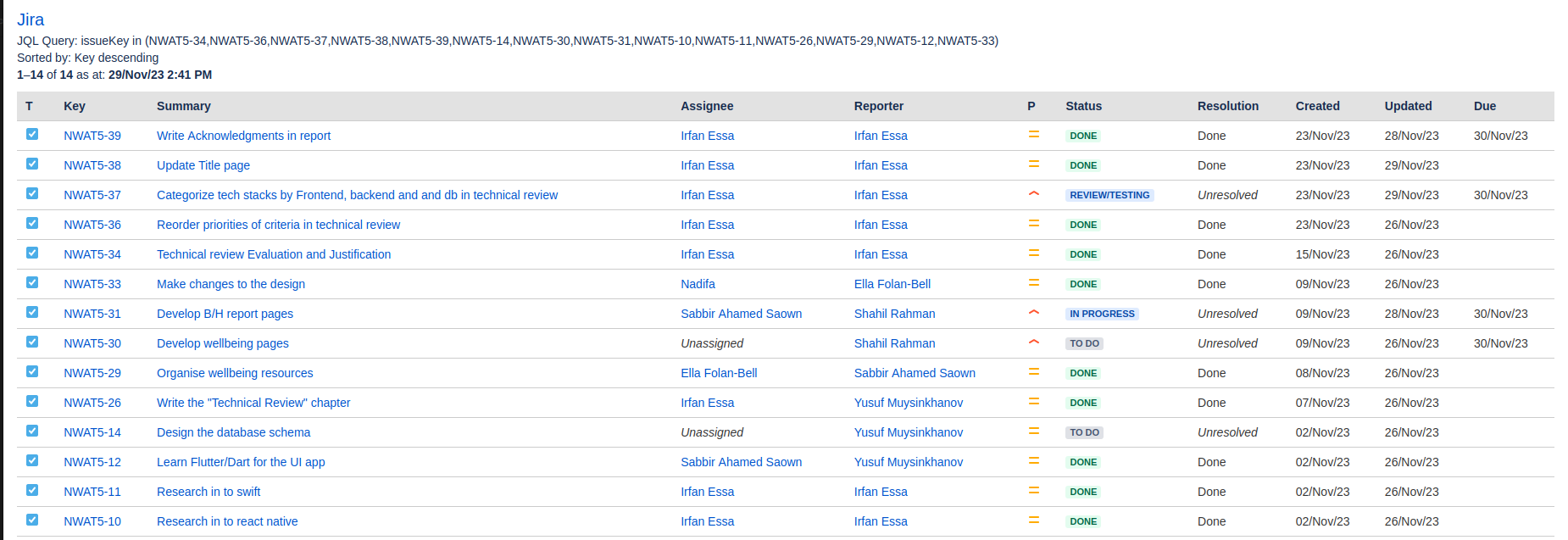
Figure 19: User Story Snippet

## **7.3. Sprints**

### **7.3.1. Sprint 1**

 Figure 20: Sprint One Snippet

### **7.3.2. Sprint 2**

 Figure 21: Sprint Two Snippet

### **7.3.3. Sprint 3**

 Figure 22: Sprint Three Snippet

### **7.3.4. Sprint 4**

A screenshot of a computer

Description automatically generated

Figure 23: Sprint Four Snippet

# **8. Retrospective meetings**

## **8.1. Meeting – Sprint 1**

A screenshot of a chat

Description automatically generated

Figure 24: Sprint one retrospective meeting review

## **8.2. Meeting – Sprint 2**

A screenshot of a computer

Description automatically generated Figure 25: Sprint two retrospective meeting review

## **8.3. Meeting – Sprint 3**

A screenshot of a chat

Description automatically generated

Figure 26: Sprint three retrospective meeting review

## **8.4. Meeting – Sprint 4**

A screenshot of a chat

Description automatically generated

Figure 27: Sprint four retrospective meeting review

# **10. References**

[1] Bernat, N. (2023). *Flutter Pros and Cons 2023 - Summary and Recommendations*. [online] leancode.co. Available at: <https://leancode.co/blog/flutter-pros-and-cons-2023-summary> [Accessed 2 Nov. 2023].

[2] Bachelder, A. (2020). 'Accelerating App Development: The Impact of Flutter’s Hot Reload'. The Developer's Digest.

[3] Wightwick, A. (2021). ‘Dart’s Compilation Advantage for Flutter Development’. Tech Times Journal, 32(4), pp. 18-23.

[4] Holmes, B. (2018). 'Streamlining Server-Side Development: Node.js with Express'. Web Dev Journal, 2(3), 45-50.

[5] Anubhav Sharma. (2023). Express JS Tutorial. [Online]. simplilearn. Last Updated: 5 October 2023. Available at: <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-express-js> [Accessed 8 November 2023].

[6] Obe, R.O. & Hsu, L.S. (2021). PostgreSQL: Up and Running. 3rd ed. Sebastopol: O'Reilly Media.

[7] Patrycja Paterska. (2023). What is React Native And When to Use It For Your App in 2023. [Online]. elpassion. Last Updated: 25 January 2023. Available at: https://www.elpassion.com/blog/what-is-react-native-and-when-to-use-it [Accessed 8 November 2023].

[8] Stack Overflow. (2023). Stack Overflow Developer Survey 2023. [online] Available at: <https://survey.stackoverflow.co/2023/#technology-most-popular-technologies>. [Accessed 8 November 2023].

‌[9] Johnson, K. (2020). 'Evaluating Swift: Performance and Efficiency in iOS Development'. iOS Development Quarterly, 12(1), pp. 22-29.

[10] MongoDB, Inc. (2023). \*MongoDB\*. Available at: https://www.mongodb.com/ [Accessed: 27 November 2023]

[11] Banker, K. (2021). MongoDB in Action, 3rd ed. Manning Publications.

[12] JetBrains (2021). Kotlin Documentation. Available at: https://kotlinlang.org/docs/home.html [Accessed: 27 November 2023]

**11. AI Statement**  
A document with text on it

Description automatically generated

Figure 28: AI declaration