P05 - Development of a Concussion Assessment and Management Mobile App

Client: Faculty of Medicine and Health - Rhonda Orr

SOFT3888_W12_01

Tutor: Islam Omar Sulieman Alzoubi

Group members:

Raymond Ton
Mariam Patel
Natalie Lu
Matthew Karko
Yupeng Zhang
Henriette Elise Onarheim
Dylan Williams (also in ISYS3888_TU15_01 doing P54)

Submission Date: 05/09/22

THE UNIVERSITY OF SYDNEY

School of Information Technologies
Faculty of Engineering & IT

ASSIGNMENT/PROJECT COVERSHEET - GROUP ASSESSMENT

Unit of Study: SOFT3888

Assignment name: First Group Report

Tutorial time: Wednesday 12-2pm **Tutor name:** Islam Omar Sulieman Alzoubi

DECLARATION

We the undersigned declare that we have read and understood the *University of Sydney Academic Dishonesty and Plagiarism in Coursework Policy*, an, and except where specifically acknowledged, the work contained in this assignment/project is our own work, and has not been copied from other sources or been previously submitted for award or assessment.

We understand that failure to comply with the *Academic Dishonesty and Plagiarism in Coursework Policy* can lead to severe penalties as outlined under Chapter 8 of the *University of Sydney By-Law 1999* (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgement from other sources, including published works, the internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

We realise that we may be asked to identify those portions of the work contributed by each of us and required to demonstrate our individual knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

Project team members				
Student name	**Student ID**	**Participated**	**Agree to share**	**Signature**
1. Matthew Karko	500493791	Yes	Yes	M.K.
2. Natalie Lu	500518599	Yes	Yes	N.L.
3. Yupeng Zhang	500049129	Yes	Yes	Y.Z.
4. Dylan Williams	490410390	Yes	Yes	D.W.
5. Mariam Patel	500494293	Yes	Yes	M.P.
6. Raymond Ton	500489473	Yes	Yes	R.T.
7. Henriette Onarheim	470053090	Yes	Yes	H.O.
8.				
9.				
10.				

SIT Building, J12
The University of Sydney
NSW 2006 Australia

T +61 2 9351 3423
F +61 2 9351 3838
E sit.info@sydney.edu.au
sydney.edu.au/it

ABN 15 211 513 464
CRICOS 00026A

Executive summary

Project title: Development of a Concussion Assessment and Management Mobile App

Group members:

- Dylan Williams
- Raymond Ton
- Yupeng Zhang
- Henriette Onarheim
- Mariam Patel
- Natalie Lu
- Matthew Karko

Project goals: develop an application that is able to detect whether or not a person has a concussion

Description of system and major achievements: the system consists of a number of tests that measure the user's reaction time, balance and ability to memorise items. The user is able to record their daily symptoms, and save all test results on their account. The application also contains a Concussion Action Plan, which provides the user with information regarding how to monitor symptoms and when to seek medical attention.

Acknowledgements: Our clients, Rhonda Orr and Andrew Fyffe

Table of Contents

1 Introduction…… 6

1.1 Problem Statement…… 6

1.2 Overall Project Vision, Goal, Purpose and Objective…….. 6

1.3 What the Project Will Achieve……. 6

1.4 Key Stakeholders…….. 6

1.5 Identification of Resources and Risks…… 7

2 Overview of System……. 7

2.1 Overview from User View…… 7

3 Evaluation…….. 12

3.1 Overview…… 12

3.2 Details of Test…… 13

3.2.1 Examples of Unit Tests…….. 13

3.2.2 Acceptance Testing…… 15

3.2.3 Usability Testing…….. 17

3.3 Conclusions……... 18

4 System Structure Overview…… 18

5 Tools to Build System…….. 19

5.1 Tools…… 19

5.2 Frameworks and Languages…….. 20

6 Information Search/Research and Discipline Knowledge Use and Application…. 20

6.1 Extreme Programming…… 20

6.2 Software Development Tools…….. 20

6.3 Data Management…… 21

6.4 Research…… 21

6.5 Information: Technical…….. 21

6.6 Current Solutions…….. 21

7 Group Processes, Reflections and Conclusions……. 22

7.1 Challenge/Risk Analysis…….. 22

7.2 Limitations in Terms of Functionality……. 22
7.3 Limitations in Terms of Structure, Design and Implementation……..… 23
7.4 Primary Strengths……. 23
8 Individual Contributions……... 25
9 References……..…….. 27
10 Appendices…… 28
A1 - User Stories……. 29
A2 - Unit Testing Summary…… 29
A3 - Acceptance Testing…… 30
A4 - Usability Testing…….. 32
A5 - System Structure Overview Diagrams…… 34
A5.1 Previous System Architecture Design…… 34
A5.2 High-Level UML Diagram for Current System Structure……. 35
A5.3 Detailed UML Diagram for Currency System Structure…….. 35

1 Introduction

1.1 Problem Statement

The problem consists of creating an application that indicates to the user whether they have a concussion based on their symptoms. The app goes through a series of symptom checklists, preliminary tests (memory, balance and reaction tests) as well as VOMS tests to check if the user has any symptoms of a concussion. The user should also be able to sign up or login and save their reports to their profile. The user should also be able to login everyday and track their symptoms after the concussion, and be advised to take the correct steps.

1.2 The overall project vision, goal, purpose or objective

The project goal is to make this application user friendly and ensure the users can conduct these concussion tests with accuracy. The user should also be able to track their symptoms everyday and save these reports under their profile. One of the objectives for this application is to ensure the usability goals are achieved as well as all the features that the client wants are easily accessible to the user when needed. An objective that the developers have also defined, is to refactor the codebase given to us as much as possible to meet code design principles and goals so that the code can be extendable in the future and be less prone to bugs. Another goal of this project is to conduct thorough testing before deployment, including unit, integration and beta testing.

1.3 What the project will achieve

This project will be ready for use by patients by the end of the sprints where they are able to input their symptoms and check whether they have a concussion. This project will also add extra features that were instructed by the clients such as Hope test, creating/sending reports to the users and being able to login and track symptoms daily. This project will also ensure that the current concussion tests are more accurate. Ultimately, by the end of this project, this application will be ready to be used by the client's patients.

1.4 The key stakeholders, what do they do, and how they interact

The clients are one of the key stakeholders for this project as they instruct the developers on the features they need implemented and provide feedback on the state of the application. The clients will be providing this app to their patients (end-users) after deployment. The clients for this project include the Faculty of Medicine, as well as, doctors in well known clients that work with concussed patients.

The development group is a key stakeholder in the project as they will be developing and deploying the application. The development group interacts through communication platforms to discuss the agendas for the sprints, assign tasks/ roles and discuss any issues in development and how to resolve them. They also interact in the weekly tutorials and on zoom to share the tasks they have completed or discuss the next steps in the sprint depending on their roles in the XP guide. The development team interacts with the other stakeholder, the clients, to work on the project.

The end-users are the key stakeholders as they will be ones using this application. They will be using this app if they have/had a concussion they would like to check or track symptoms for. The end-users could also be people that conduct the concussion test on others (coach of the team, parent) therefore, these users will interact with one another either by sharing the concussion reports to each other or through verbal communication. These stakeholders are an important aspect in development as the app needs to be user-friendly based on these user's needs. During beta testing, the development team or the clients will be interacting with some of these end-users to provide feedback on the application.

1.5 Identification of resources and risks in this project

The resources provided to the group have been studies and reports by the clients on background information about the features being implemented (concussion action plan provided for that feature to be implemented on app). As this app implements concussion tests, the clients also have provided the group with steps to do the tests and what the data would indicate (passed or failed the test).

The risks involved in this project include inaccuracy of the tests leading to individuals being misdiagnosed with their symptoms, therefore, the features need to be implemented with the most accuracy and need to be tested. Another risk involved with this project is that the app is unable to be deployed due to the lack of a server provided store the data from the app. This can be a big issue once the app has been deployed as users have a chance of losing their data. inability to time manage and complete the project in time is a huge risk, therefore, the group has set up sprints to prioritise the important features to be implemented first, then the features that the clients aren't as eager to have implemented.

2 Overview of system from user view (user stories)

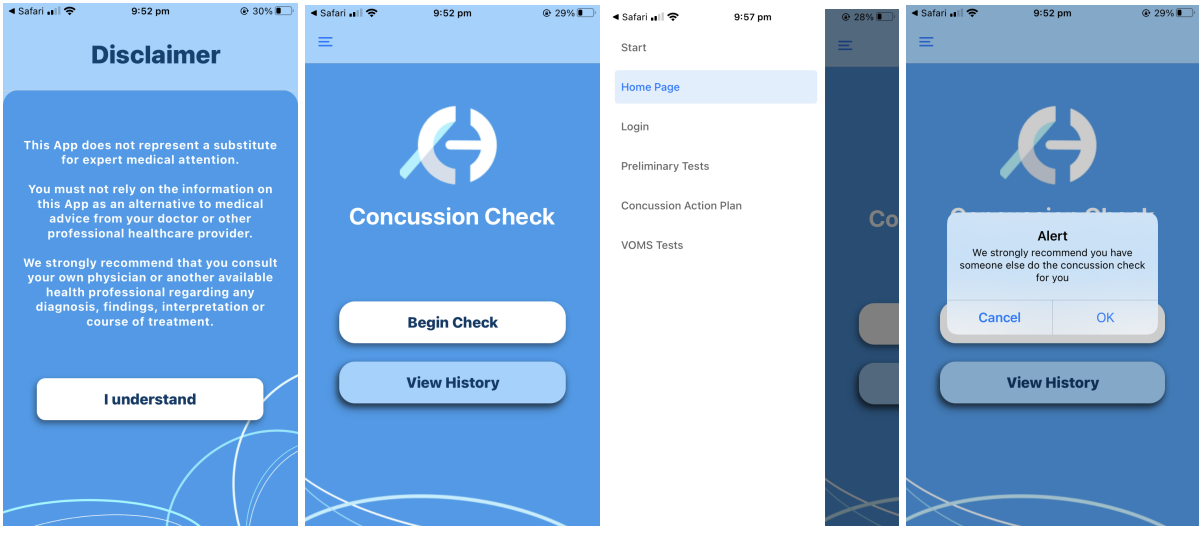

Figure 2.1-2.4. Opening of application and menu bar

The above screenshots show what the user sees upon starting the application. The user will first be met with a disclaimer letting the user know that the app is not a substitute for expert medical attention and should only be used as an indication to

whether or not the user has a concussion. After clicking on “I understand”, the user will be met with the home page which is shown in the second photo. Clicking on the menu bar icon in the top left corner will give you the menu shown in the third photo, while the fourth photo shows the alert message displayed when the user clicks on “Begin Check” on the home page. The same alert message would come up if the user clicks on “Preliminary tests”.

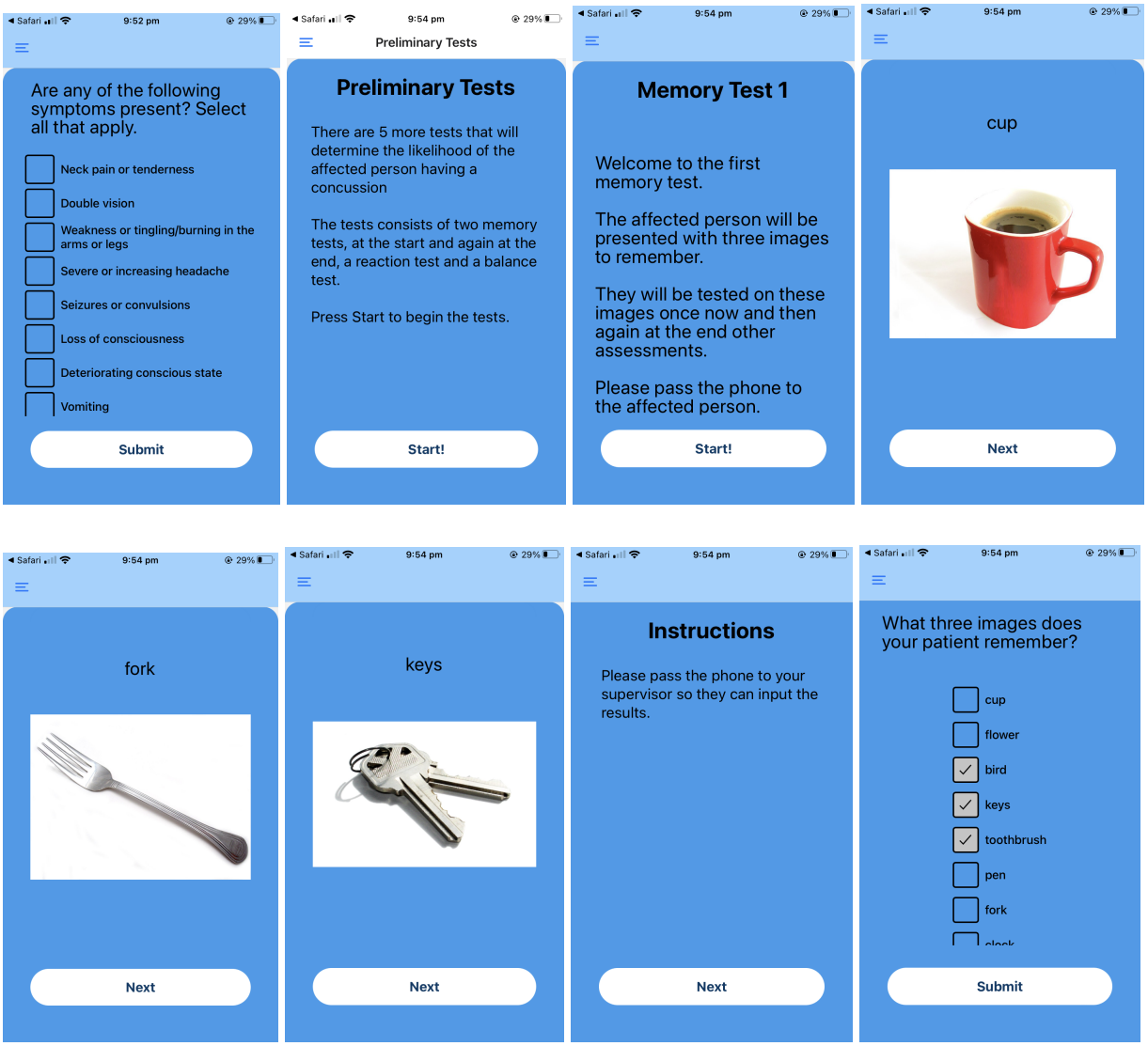

Figure 2.5-2.12. Symptoms log and first memory test

The above 8 screenshots are placed in order of what is displayed after the user has pressed “Begin Check”. The user will first be able to use the symptoms log, before moving on to preliminary tests where a memory test is the first one being conducted. Three random images will be displayed, and the user will need to remember the items displayed on the images before being tested.

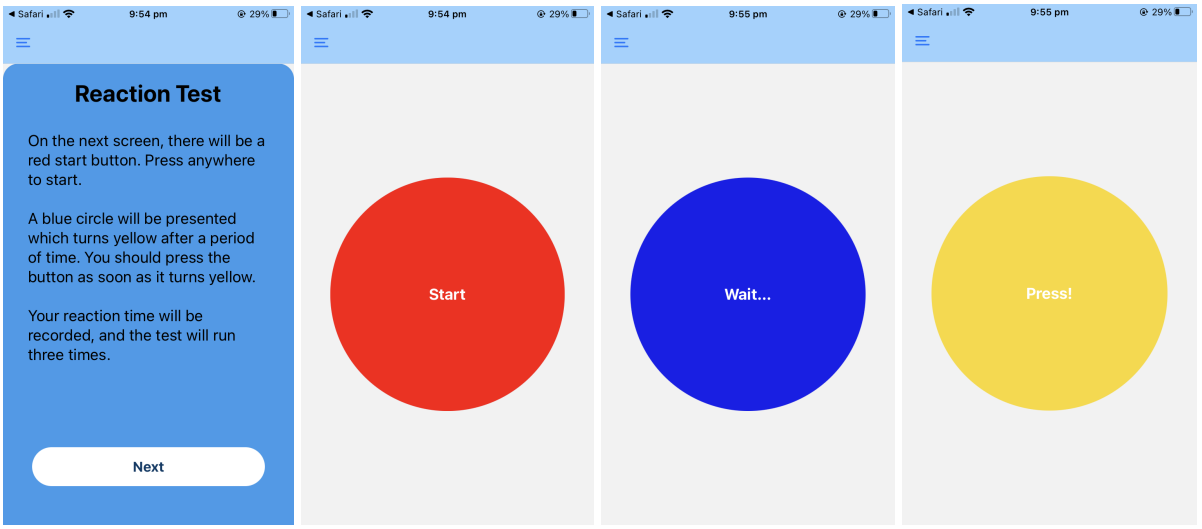

Figure 2.13-2.16. Reaction test

The above four screenshots show the reaction test which follows right after the first memory test. A red circle will be displayed, which the user must press. The circle will then turn blue, and after 2-5 seconds the circle will turn yellow. As soon as it turns yellow, the user will need to press it again in order to get to the next page.

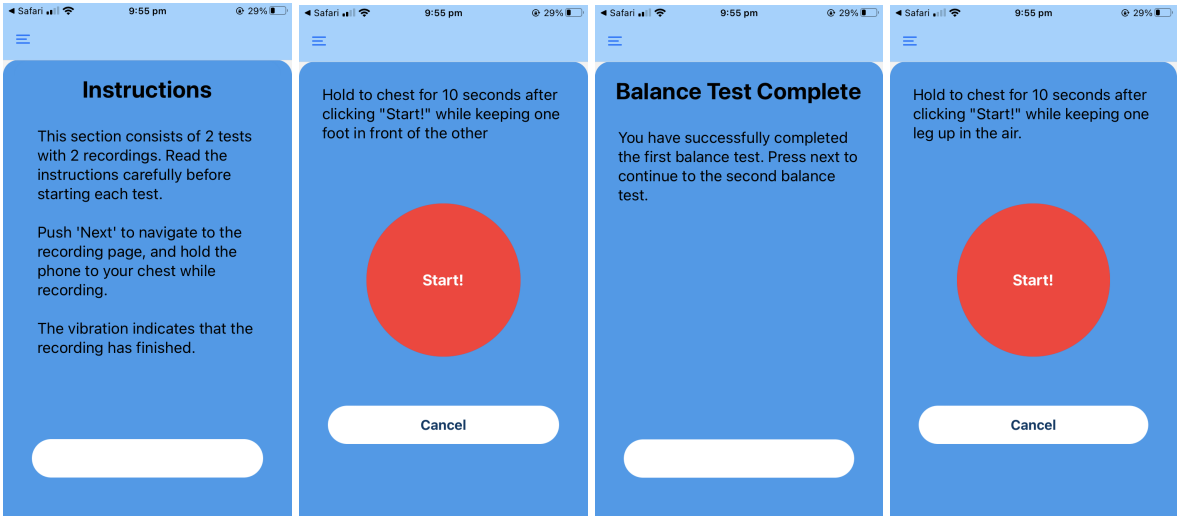

Figure 2.17-2.20. Balance test

After the reactions tests are completed, the user will be taken to the balance tests which first tells the user to keep one foot in front of the other for 10 seconds, and then keep one leg up in the air for 10 seconds. Please note that for image 2.17 and 2.19, the white button does not say “next” on it. This is a bug that we are currently working on resolving.

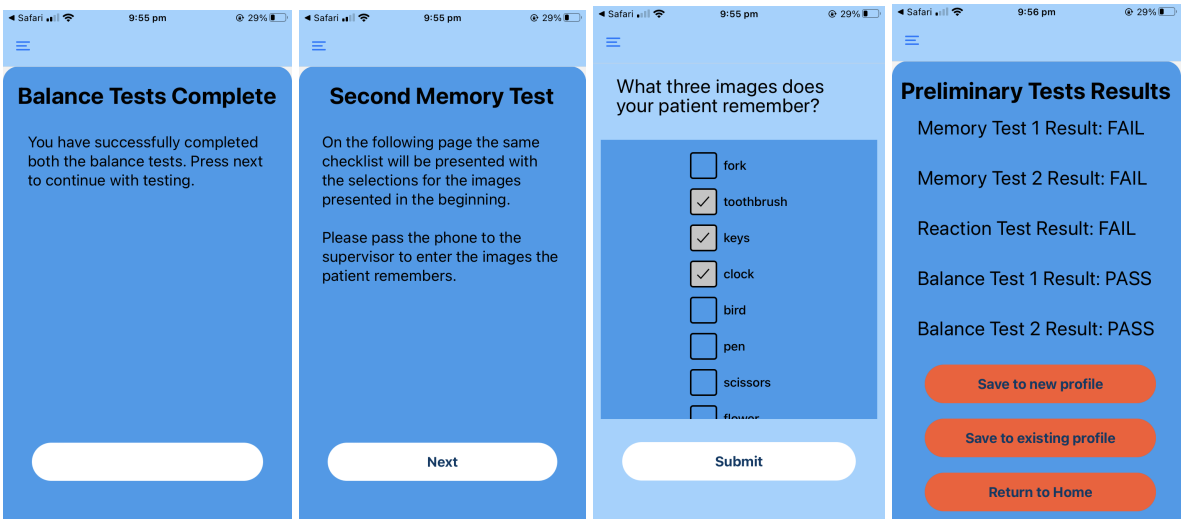

Figure 2.21-2.24. Second memory test and test results

After the balance tests are completed, the user will need to do a final memory test before receiving the test results. This time, the user will not be shown images but will rather need to remember which items were displayed earlier. After clicking on the “Submit” button in image 2.23, the user will be shown test results which may be saved to either a new profile or an existing profile.

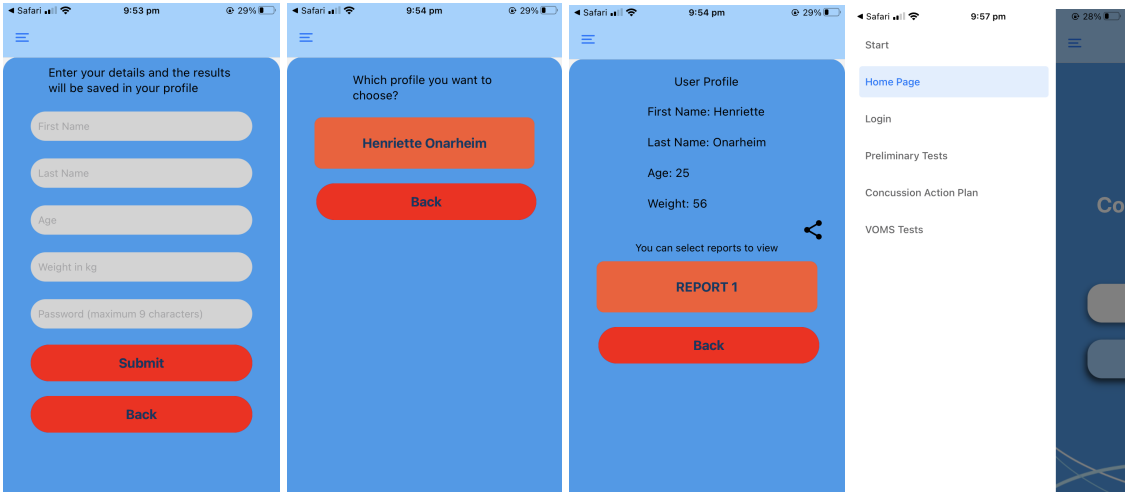

Figure 2.25-2.28. Creating a profile and viewing existing profiles

The first image above (figure 2.25) shows what happens if the user presses the “Save to new profile” button after receiving the test results in figure 2.24. The user needs to provide some personal details and create a password for the profile, which can be used to log in to the profile next time the user is using the application.

Figure 2.26 shows what happens if the user presses the “Save to existing profile” button after receiving the test results in figure 2.24. This page can also be accessed from the side menu by clicking on “Login”. If the user clicks on a user, such as “Henriette Onarheim”, all user information will be displayed and the user will also be able to select a report that has been saved to the account.

After creating a new profile or viewing an existing profile, the user is able to navigate to

other parts of the application by clicking on the top left menu icon which displays the side menu shown in figure 2.28.

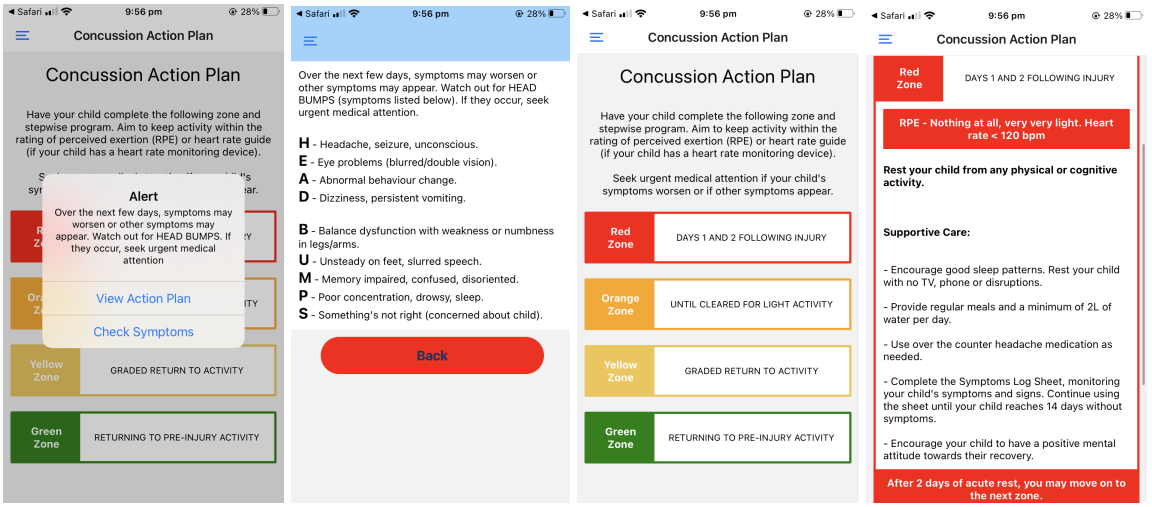

Figure 2.29-2.32. Concussion Action Plan

If the user clicks on “Concussion Action Plan” in the side menu, he will be taken to a page which displays information regarding symptoms and how to monitor them. As shown in figure 2.29, the user will first get the option to “Check Symptoms” or “View Action Plan”. Clicking on “Check Symptoms” will take the user to figure 2.30, which displays information regarding HEAD BUMPS symptoms. Clicking on “View Action Plan” would take the user to figure 2.31, which displays information regarding how to care for the concussed child depending on the severity of the symptoms.

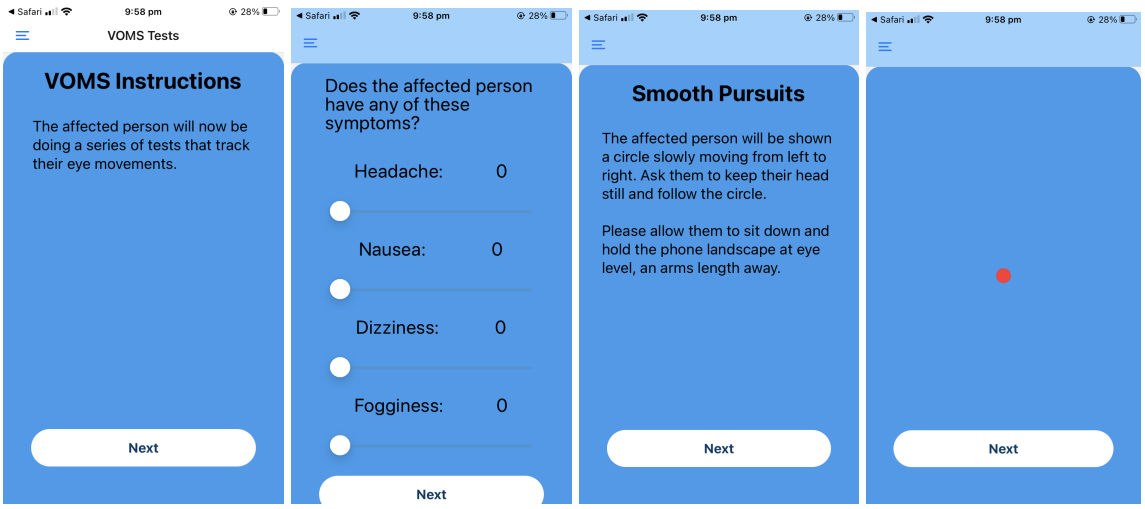

Figure 2.33-2.36. VOMS Testing

From the menu bar, the user may click on “VOMS Tests” which takes the user to figure 2.33. From there, the user will need to go through a number of tests such as smooth pursuits and recording symptoms. However, this part of the application is not yet finalized.

Please note that our application does not allow ‘admin’ users, hence why we have only included user stories from a normal user’s point of view.

3 Evaluation

3.1 Overview

Current requirements and acceptance criteria:

- Users can create a profile, and may choose to save test results to the profile such that they can be accessed later.
    - The acceptance criteria for this one is that a profile is successfully created and may be accessed even after closing the application. Test results will be saved on the profile, however creating a profile should not be necessary in order to perform any tests. This is because it is a medical app, and some users may be under time pressure and do not have time to sign up before performing tests.
- User should be able to share results with others in some way, for example as an excel spreadsheet or a text file
    - Although the exact way of transferring data has not yet been decided, the current acceptance criteria is that the user can send the results as a text file
- Colours used in the application must be distinguishable, otherwise this may cause confusion for colour blinded users
    - Acceptance criteria here is that red and green are not used within the same page. The client has expressed that they want us to use mostly blue colours
- A Concussion Action Plan should be available from the menu, which includes all necessary information regarding how to monitor symptoms
    - Acceptance criteria here is that the Concussion Action Plan is easily accessible and contains correct information regarding symptoms, how to monitor symptoms etc. The client has provided us with this information.
- User is able to record symptoms daily
    - Acceptance criteria is that the user can record symptoms daily, without having to go through every single preliminary test all over again. The symptoms should be saved to the user's profile if they are logged in.
- A memory test which allows the user to see three images, before having to memorise them and select them from a checklist. After performing a reaction test, the user will need to select from the same checklist again, and
    - Acceptance criteria is that all three images are different, and they should be displayed on the screen until the user presses "next". The app should only give the user a "pass" grade if the user correctly identifies all three images.
- A reaction test which allows the user to see a blue circle, and as soon as it changes colour the user will need to press the circle as quickly as possible. The reaction time is recorded
    - Acceptance criteria is that the user is able to click on the blue circle, as well as clicking on the same circle when it turns yellow. If the reaction

time is less than 500 milliseconds for all three tests, the grade will be “pass”.

- A balance test which allows the user to hold the device close to his chest for 10 seconds while keeping one foot in front of the other. There will be a second balance test right after where the user will keep his foot up in the air for 10 seconds.
    - Acceptance criteria is that the user is able to perform these balance tests and that the result correctly identifies if the user has poor balance and possible cerebellar damage

These requirements are also expressed as user stories on Wiki [1].

3.2 Details of Test

So far in this project, we have mostly been focusing on unit, accepting and usability tests to ensure that each part of the program works as intended and that the application meets all client requirements. Due to that we were given a codebase with relatively poor quality, we had to refactor a large part of the code which could potentially lead to bugs. Unit testing has allowed us to ensure that we do not break any code when refactoring, and that the program is still working.

Examples of unit tests in our program

- Testing to see if the reaction test functions behave correctly. For these tests, we have used Jest matchers (such as the **expect** function) for validation, as well as mocks to isolate the code being tested. The example has been taken from ReactionTest.test.js [2].

```
71   describe('getReactionTest', () => {
72     it('returns existing results', async () => {
73       const mockRs = {
74         rows: {
75           length: 1,
76           item: jest.fn(() => TEST_REACTION_RESULTS),
77         },
78       };
79       mockDa.runSqlStmt = jest.fn(() => Promise.resolve(mockRs));
80       let r = await rt.getReactionTest(TEST_REACTION_RESULTS.rt_id);
81
82       expect(r.rt_id).toBe(TEST_REACTION_RESULTS.rt_id);
83       expect(r.time_attempt_1).toBe(TEST_REACTION_RESULTS.time_attempt_1);
84       expect(r.time_attempt_2).toBe(TEST_REACTION_RESULTS.time_attempt_2);
85       expect(r.time_attempt_3).toBe(TEST_REACTION_RESULTS.time_attempt_3);
86       expect(r.time_average).toBe(TEST_REACTION_RESULTS.time_average);
87     });
88   });
89 });
```

Figure 3.2.1. Reaction Test Testing

- Testing if we are able to correctly get the standard deviation of an array. We are using the **expect** matchers for these unit tests as well, to ensure that the expected result equals the actual result. For example, if the given array is empty, the function should return 0. The example has been taken from standardDeviation.test.js [3].

```
describe('standardDeviation', () => {
  it('returns 0 when given empty array', () => {
    const arr = [];
    const std = standardDeviation(arr);

    expect(std).toBe(0);
  });

  it('returns standard deviation is not 0', () => {
    const arr = [1, 2, 3];
    const std = standardDeviation(arr);

    expect(std).toBeCloseTo(0.81649);
  });
```

Figure 3.2.2. Standard Deviation Testing

- Testing to see if the application is able to create a new profile. As the other examples above, the unit tests for this part of the program also use mocks and **expect** matchers in order to validate that a profile has been created successfully. The example has been taken from PatientRepo.test.js [4].

```
  describe('createPatient', () => {
    it('calls runSqlStmt with correct args', async () => {
      const fN = 'john';
      const lN = 'smith';
      const age = 34;
      const weight = 69;

      await pr.createPatient(fN, lN, age, weight);

      expect(mockDa.runSqlStmt.mock.calls.length).toBe(1);
      expect(mockDa.runSqlStmt.mock.calls[0][1]).toEqual([fN, lN, age, weight]);
    });
```

Figure 3.2.3. Creating a Profile Testing

The examples provided above are only a small number of unit tests that we have in our program, and a larger number of tests can be found in the __test__ folder [5]. As shown above, we have used a variety of testing techniques with our unit tests, such as white box testing and black box testing. Although we have already created a large number of test cases, we are aiming to create even more in the coming weeks to ensure that the application does not contain any bugs or crashes unexpectedly.

As mentioned in the examples above, we are using Jest in order to test our program and by running **npm run ci-test** in the terminal we get a detailed overview of the code coverage.

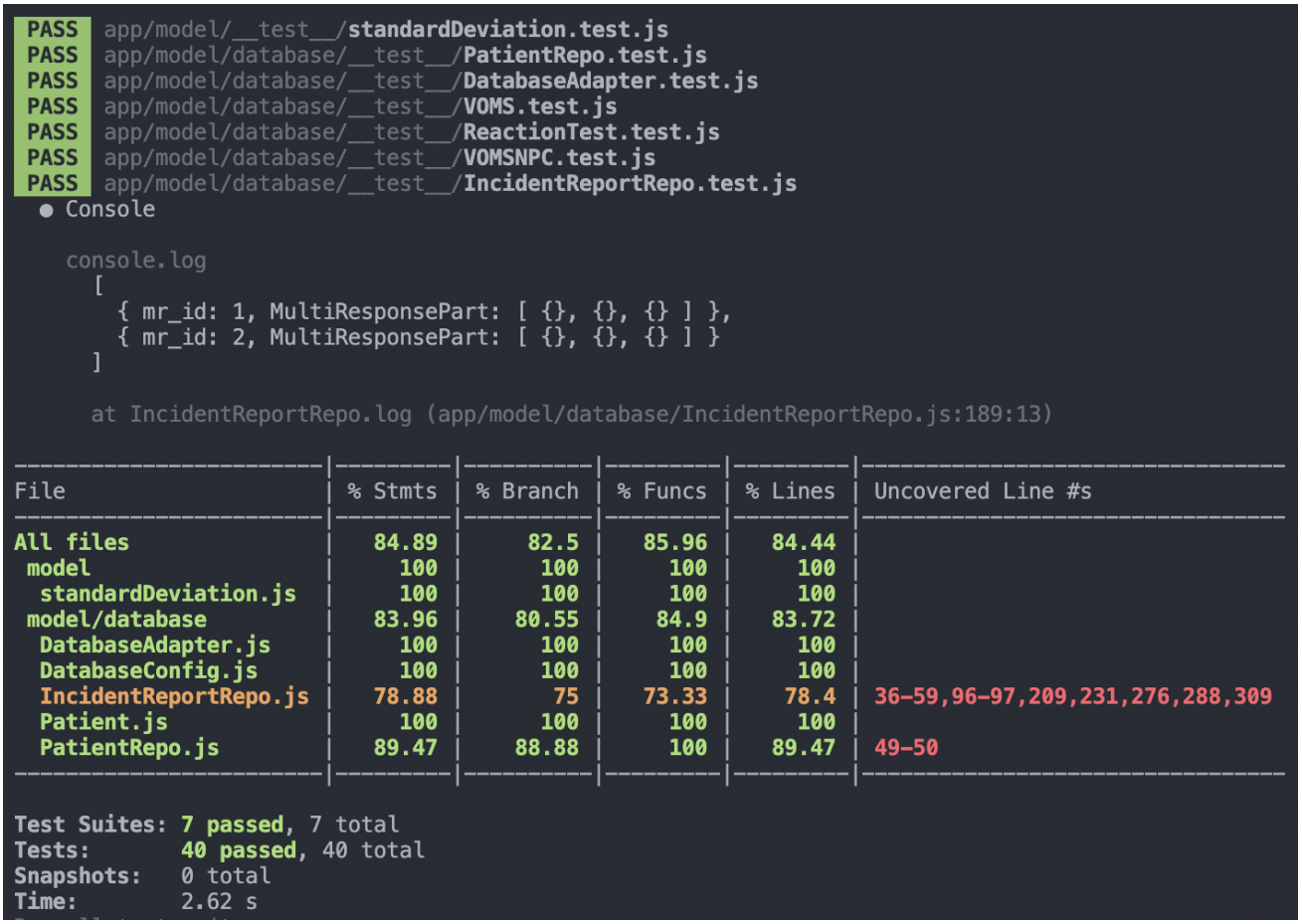

Figure 3.2.4 Code Coverage

As shown in the screenshot above, the code coverage for the program is relatively high and we have covered all of the essential parts of the program. All of the test cases are passing, which indicates that the program is working as intended. The majority of them are testing functionalities of the program that are related to our user stories [1], such as “I want to perform the reaction test” and “I want to create a profile” which have been tested in the ReactionTest.test.js and PatientRepo.test.js files. All parts of the program that are related to the user stories have been tested with unit tests, however we have also used acceptance testing and usability testing to ensure that the users are able to perform all tasks they want to do on the application.

Acceptance Testing

For user acceptance testing, we have asked two participants to perform a number of tasks on the application while we are observing. The scope of the testing was to conduct all preliminary tests, view the concussion action plan and go through the PCSS checklist. The VOMS tests were not conducted due to that these tests are not finalized yet.

6. UAT Requirements-Based Test Cases			
ID	**Test Cases**	**Criteria**	**Results**
6.1	Preliminary tests: • Open the app on iPhone • Select “Preliminary tests” from the menu bar	Expected result: • This tester finishes all the tests	Participant 1: PASS Participant 2: PASS

	• Follow the instructions on memory, reaction, and balance tests	• The results of all tests are presented to the user	
6.2	Red flags & PCSS Checklist: • Open the app on iPhone • Click “Begin Check” button on the main page • Click “Ok” button on the pop-up alert window • Select symptoms from the list of checkboxes (randomly selected, since tester does not have a real concussion) • Click “Submit” button on the bottom of the page • Click “Save to new profile” on the result page after the checklists • Type in personal details and click “Submit” button • Click “View history” button on the home page to view the result	Expected result: • Personal profile successfully created • All selected symptoms are saved, viewable in profile and following instructions for the use are prompted	Participant 1: PASS Participant 2: PASS
6.3	Concussion Action Plan • Open the app on iPhone emulator • Select “Concussion Action Plan” from the menu bar • Click “Check Symptoms” button on the pop-up alert window • Have a read through the head bumps information on the page • Click “Back” button on the bottom of the page • Select “Concussion Action Plan” from the menu bar • Click “View action plan” button on the pop-up alert window • Have a read through the information of each zone on the page	Expected result: • The alert window successfully popped up • All the info on the Head bumps and action plan pages are clear and viewable	Participant 1: PASS Participant 2: PASS

Table 3.2.5 User Acceptance Requirements

Please note that in order for a participant to receive a PASS for the test cases, the participant needs to successfully perform each of the tasks provided in the “Acceptance Tasks” column and meet the criteria for the given row.

A summary of the details of the acceptance tests can be found in the appendix at A4 - Acceptance Testing.

Usability Testing

Due to that the project is still in its early stages, we have only performed informal usability testing on a small number of participants for now. We asked the participants a number of questions regarding the features and navigation of the application, and observed their reactions. Details about the scope and participants can be found in the appendix at A5 - Usability Testing.

Usability Testing			
ID	**Tasks/Questions**	**Observations (ID: 1)**	**Observations (ID: 2)**
1	If you wanted to create a profile, where would you go?	From the home page, the user pressed the menu icon in the top left corner followed by “Login”. The participant then entered her details and clicked on “Submit” which gave an error message. The participant then clicked on “Create Login”, and filled in details again and pressed “Submit”.	From the home page, the user pressed the menu icon in the top left corner followed by “Login”. The participant then clicked on “Create Login”, filled in her details and pressed “Submit”
2	Imagine if you felt nauseous and wanted to check if you have a concussion. How would you do this?	The participant clicked on “Preliminary tests” from the side menu, and performed all of the tests.	The participant clicked on “Preliminary tests” from the side menu, however she did mention that it was difficult to know if she was meant to press “VOMS Tests” or “Preliminary tests” due to that she does not have a lot of knowledge about concussions, hence she did not know what VOMS tests are.
3	For each preliminary test (memory, reaction and balance), how would you start the tests?	For the memory tests, the participant pressed the white “Start button”. For the reaction tests, the participant pressed the red circle. For the balance tests, the participant pressed the red circle.	For the memory tests, the participant pressed the white “Start button”. For the reaction tests, the participant pressed the red circle. For the balance tests, the participant pressed the red circle.
4	If you wanted to have a look at the Concussion Action Plan, how would you do this?	The participant clicked on “Concussion Action Plan” from the side menu, and was able to successfully access the action plan	The participant clicked on “Concussion Action Plan” from the side menu, and was able to successfully access the action plan

Table 3.2.6 Usability Testing

Summary of Observations	
ID	**Observation**
1	It seems like the participants may have been confused by how to create a profile. For example after clicking on “Login”, the “Submit” and “Create login” button seem fairly similar and it is not clear whether or not details need to be filled in before pressing “Create login”. Participant 1 had to fill in details twice because there were confusions around this, so a suggestion would be to change this part of the application to make it more clear how to login and create a new profile.
2	One of the participants mentioned that it was difficult to know the difference between preliminary tests and VOMS tests due to that she does not have any prior knowledge of any of this. A suggestion would be to include some more information about VOMS tests, and the difference between preliminary and VOMS.
3	It seems like both participants were able to successfully start the preliminary, which indicates that this part of the application has a high usability and is easy to understand
4	As for the previous task, it seems like the participants did not have any issue finding the concussion action plan. This indicates that it is accessible within the app and relatively easy to find.

Table 3.2.5 Summary of Observations - Usability Testing

While we have currently only performed acceptance testing and usability testing on four participants, we are aiming to conduct more formal testing on a larger number of participants in the coming weeks.

3.3 Conclusion

After performing the acceptance and usability testing, we have gained some insight into how the application is perceived by someone who has never seen it before. The participants were able to complete the majority of the tasks given to them, which indicates that the application is relatively usable. However, there are certain areas that may need improvement, such as the login page and the menu bar. This is something we will aim to improve in the following weeks.

We will also aim to increase the code coverage further and create more unit and integration tests. We will follow the test-first approach, and ensure that all code pushed to the master branch is fully tested before it is merged.

4 System structure overview

The whole project is about taking over and fixing the codebase issues left by last year's students and continuing to implement new features into the current codebase. A system architecture overview diagram before the changes provided by the previous group can be found in A7.1 – Previous system structure overview.

To better understand the current system structure overview, the group designed UML diagrams to represent the system's overall architecture in both high-level and detailed

diagrams. Here is the list of essential components in the system:

- **Screens package:** contains all the GUIs which render the application screen on hardware based on data from the model. This package mainly consists of all the pages of the preliminary tests, VOMS test, concussion action plan, and head bumps.
- **Components package:** contains all the logic and details of UI design, including text alignment, buttons and checkboxes' press function and layout.
- **Assets package:** contains all the external resources like pictures and beep sound for the tests in the app.
- **App.js:** contains all the pages' navigation logic.

Diagrams of the system structure overview at the current stage can be found in A7.2 and A7.3. The high-level diagram visualizes the system's relationships and interactions of different packages. The detailed diagram includes a more specific design (to files-level) of the system and has labelled all the changes made by the current team. The changes labelled in blue indicate the package or files are developed or changed by the team to provide a better usability design and workflow, like improving preliminary tests' usability. The changes labelled in red represent the files or packages that are newly added by the team for implementing new features required by the client of this project.

5 Tools to build system

5.1 Tools

1. React Native
    a. React native was used since it was part of the existing application, and allows cross-platform mobile application development with a singular codebase.
2. SQLite
    a. Used for storing information in a relational structure regarding patient data and test results. This allows for easy and structured access
3. Jest
    a. Used for unit testing database class and other react components. This integrates with the BitBucket build pipeline to ensure pull requests do not break existing functionality (regression testing)
4. Expo
    a. Used for managing the React Native codebase and dependencies for Xcode on IOS and Gradle on Android.
    b. Allows us to easily build the application .apk and .app files for mobile devices on the expo server without the hassle of setting up an isolated build environment.
5. Git and Bitbucket
    a. Used to keep track of the version history.
    b. XP programming is used with code reviews and continuous integration. Each time a commit to the master branch is created via a pull request, the build pipeline will automatically build the app for IOS and Android

and download the files into the repository.

5.2 Frameworks and Languages

1. JavaScript
    a. This language is the primary language of the app. All components, and database logic is written in javascript.
2. React
    a. React is a UI framework developed by Facebook [6]. It is designed to compartmentalise UI components in a modular and reusable way.
3. React-Navigation
    a. This is a library used for adding navigational functionalities to the app. This was mostly used for creating the global menu to quickly access key pages such as the Action Plan and preliminary tests.

6 Information search/research and discipline knowledge use and application

6.1 Extreme Programming

Throughout the development of our project, we have closely adhered to the agile software development framework, Extreme Programming. We have utilised a fortnightly sprint to implement and improve features of the system. At the start of the sprint, user stories are made into tickets and then delegated between the development team. During the sprint cycle, unassigned tickets are allocated to team members that completed their ticket. A meeting is held with the client every week to deliver the progress on the implementation. With the weekly communication, the team continually receives feedback and suggestions from the client to formulate new user stories for the sprints. Through advocating extreme programming, we have proactively adapted the application to satisfy the client requirements.

Alongside the Extreme Programming methodology, each member was assigned a role. Every team member was a programmer to maximise productivity. Matthew was the main point of contact for the client as he was the manager. Natalie played the role of tracker to assist with recalling back to our plan. Everyone effectively followed the principles of their roles, resulting in a productive development cycle.

6.2 Software Development Tools

Expo was a platform that we used to deploy the application. During the development of features, we were able to build the app onto our physical and emulated phones to test the functionality. Through adopting this framework, it made the implementation more convenient due to the automated rebuilding.

For the project, Bitbucket was the collaboration tool that we used to host our repository. Whenever team members implemented or modified features, they would stage the changes to a new branch and create a pull request. Other members would approve of the request before it is merged. With this tool, our team was able to

implement different features of the application concurrently.

To organise our sprints, we utilised the board Jira to log the user stories to complete. Tickets were created based on the user stories and delegated to the team members. Jira managed each sprint by monitoring our progress and enabling the visibility of the status on tickets. Through using Jira, our team operated seamlessly in the sprint cycles.

6.3 Data Management

To store data in our application, we utilised the SQLite module from the package Expo to create a database. The database is initialised locally on the device once the application is run. Tables to store the results of preliminary tests are defined alongside the functions to receive the data from the table. When updating the table, the data is gathered through the interface and cached at the end of each test interval. Test suites have been written to verify the instantiation of tables and access clauses.

6.4 Research

In order for us to completely understand how to implement the solution in a manner that is extensible, maintainable and usable, background research into both the medical aspects of the application as well as the technical aspect of the application is required.

6.5 Information - technical

In this assignment we were given an existing codebase that utilised React native as well as expo which are industry standard tools. All the initial research was already completed into what technologies to use by the previous development team, however the codebase that was provided was heavily outdated and used deprecated versions of libraries as well as logic that was hardcoded. So from this our team had to research the compatible versions and update the deprecated application to a version that is compatible with current standards whilst not breaking any of the current working code.

Research into specific react native methodologies such as react-navigation types, page view types and logic was required in order to make sure the application becomes effective.

6.6 Current solutions

A part of making sure the quality of the application is meeting and above industry standards, our team did research into current successful applications in the concussion management market to understand what types of technologies they use and why they are successful.

Through research it was found that this market is very niche and there are no solutions that directly solve the issue that the client would like us to solve. However there is an application called HeadSpace which is a very basic head health checker.

The application is full of redirects and offers no real value when completing their head assessments rather links to external pages or to phone a medical professional. Although we do not have access to what types of technologies they have used to develop the application, it is clear that they have some form of server running to store user data and that they have some form of frontend framework such as React which is what we are using.

From this research it is clever to say that our project is on the correct path and that we are implementing the correct discipline knowledge.

7 Group processes, reflections and conclusions

7.1 Challenge/ Risk Analysis

Our project builds on an existing prototype concussion detection application, therefore, there is a codebase provided to us for further developments. This posed a challenge to our group as we cannot decide the language, frameworks and technologies for the project by ourselves, instead, we have to follow what has been used in the codebase which the group is not very familiar with. Therefore, before we start development, some of the team members have to self-learn the technologies they don't have experience with before and since the project is extending from an existing application, we have to review the codebase and understand their design and structure. The risk raised from it is inaccurate estimation as there are many uncertainties in the beginning stage of the project, we cannot accurately estimate the length of each task for the technologies we are unfamiliar with. Moreover, the codebase provided is not properly commented which greatly affected our understanding of the code and the process of project planning. Another challenge for our group is the uncertainty of the requirements. Our client is a professor in medicine school who has excellent professional knowledge in concussion, yet has very limited knowledge in computer science. The requirement she proposed at first is technically infeasible given our group's current skills and experience. In order to push the project forward, we had several meetings with her to alternate the requirements to meet our group's current abilities and propose new, additional deliverables. The risk raised from it is scope variations as our scope had a major change and we are still understanding and receiving new requirements which could impact our group's ability to stick to the planned timeline of the project.

7.2 Limitations in Terms of Functionality

The VOMS tests we are implementing require a 3 foot distance to detect the eye movement which is not possible on a phone screen. The alternative way we proposed is to record the eye movement and save it for a health professional to review instead of having the system to perform the check. This could extend the diagnosis time and decrease the accuracy.

The reports and user profiles are currently stored locally. To make the application work for general users, we need to have a database server as a central to the flow

and preservation of data. However, we do not have any funding for this project to purchase a database service and establish a server for a remote database. Therefore, the login, sign up and view report features will work logically but not as usable as these features should be in a released application.

Although this application can perform some preliminary diagnosis of concussion, users cannot use the results as a formal diagnosis. It can only be used as a reference and symptoms recorder, users still need to go to professional medical institutions for further inspection.

7.3 Limitations in Terms of Structure, Design, Implementation

Since our project builds on an existing codebase, we have to follow their design structure, only minimal changes in structure can be made otherwise the functionality of the application can be affected. Moreover, the provided codebase has specified the framework we should use for implementation, despite most team members not familiar with the framework, we have to follow it as we won't be able to refactor the code given the limited timeframe.

7.4 Primary Strengths

7.4.1 Reflections on Extreme Programming

The group follows the Extreme Programming practices and values. Each member has communicated actively in the Slack discussion channel. Although most of the group are not familiar with the specified technologies and framework, there are few members who have experience with them and they transfer the knowledge they have to the group through communication and in person assistance which have a great positive impact on the project progress. We communicate with clients weekly to update the project progress and ask for feedback. The group is always prepared for changes and new requirements from the client. For each change made, each feature completed, members will notify the group and create pull requests for review and feedback. This not only avoids the possible error, but also significantly improves the code quality. The group uses user story as a starting point of the implementation as we know the importance of understanding the requirements. Then we will start planning based on the stories and estimate the effort we will put in for each story. We started with the smallest yet useful feature set such as menu bar for usability, theme color for UI, fix the issues in each test and after that, we are continuously adding new features such as navigation, action plan, page design etc. The group follows the continuous integration, any changes and features are integrated into the codebase daily and the code will be checked against the tests both before and after integration.

7.4.2 Ways of applying version control, issue tracking, coding styles

The group uses Git as the version control tool and BitBucket as the Git repository management solution. The group uses Jira for issue tracking where we set up sprints and issue tickets for each sprint. Group members will assign themselves to the tickets and update the status of the ticket as “progressing” or “done” so that the group can track the project progress and individual contribution. Wiki is for recording the project information including the XP roles, meeting minutes, and individual contributions.

The group agrees on commenting on the code along with the development so that it will be easier for others to review. Before the change gets merged into master, members have to create a pull request for other members to check and approve to ensure the code quality.

7.4.3 Group aspects as well as product and processes

The group aspects are: Responsibility - where all members are expected to be responsible for their tasks and communicate once they have difficulties to complete the task; Respect - where all members should respect each other, communicate the issue to avoid conflict and encourage the others rather than criticise; Organised - where the group will have weekly meetings, make plans and follow the plan; Passionate - where each member should be passionate about the task they got assigned, self-motivate to complete the task on time and ensure the quality of the result. Communication - where all members are expected to discuss actively in the Slack group discussion channel, share their opinions and knowledge to the group and raise issues which concern them.

The product we are aiming to produce should meet all the client requirements, have a satisfying performance where it should be able to perform as a preliminary test tool for concussion. It should be easy for the first time user to use and have a pleasant user’s interface. It should be able to handle all possible user’s input and will not crash due to unexpected input or hidden bugs.

The process of production is firstly, weekly client meetings to receive and confirm the requirements, update the project progress and receive feedback. Then, the group will determine the user story based on the requirements, estimate the effort needed to be put into each story, and issue the tickets based on the stories for the current sprint. Each sprint lasts for two weeks and the group will update the progress, discuss the issues and make adjustments to the plan during weekly group meetings. Group members will also communicate through Slack for urgent issues or immediate attention. Lastly, all code will be reviewed by the group to ensure the quality.

8 Individual contributions and work split

Table of contributions

Name/Task	Natalie	Henriette	Raymond	Dylan	Mariam	Yupeng	Matthew
Setup bitbucket				✔			✔
Setup jira				✔			
Create menu					✔		
Disclaimer page UI	✔						
Home page UI	✔						
Acceptance testing						✔	
Usability testing		✔					
Finalised report and references		✔					
Improved reaction test		✔					
Wrote progress report		✔			✔		
Added documents to wiki		✔	✔				✔
Created Gantt chart		✔					
Create preliminary report storage and display page							✔
Meeting Minutes	✔						
Improved usability of the balance tests			✔				
Improved usability of the memory test						✔	

Fixed the checkbox function in memory test						✔	
Organising and facilitating all meetings							✔
Update previous year's codebase dependencies			✔				✔
Peer review	✔	✔	✔	✔	✔	✔	✔
Created Action Plan				✔			
Added HEAD BUMPS				✔			
Design Lo-Fi Prototype	✔						
Design Hi-Fi Prototype	✔						
Created/ updated login feature					✔		
Fixed storage of users in the database					✔		
Configure Bitbucket Build Pipeline				✔			

Table 8.1 Contributions

9 References

[1] R. Ton et al. “Wiki: User Stories”. Accessed: Sep. 3, 2022. [Online]. Available: https://bitbucket.org/w12-1-capstone/w12_1_capstone/wiki/User%20Stories

[2] M. Karko et al. “BitBucket: w12_1_capstone/ReactionTest.test.js”. Accessed: Sep. 2, 2022. [Online]. Available: https://bitbucket.org/w12-1-capstone/w12_1_capstone/src/50b6a6a6fa86d76d5fe4457d156f1419eefd171d/app/model/database/__test__/ReactionTest.test.js

[3] D. Williams et al. “BitBucket: w_12_1_capstone/standardDeviation.test.js”. Accessed: Sep 2, 2022. [Online]. Available: https://bitbucket.org/w12-1-capstone/w12_1_capstone/src/50b6a6a6fa86d76d5fe4457d156f1419eefd171d/app/model/__test__/standardDeviation.test.js

[4] N. Lu et al. “BitBucket: w_12_1_capstone/PatientRepo.test.js”. Accessed: Sep 1, 2022. [Online]. Available: https://bitbucket.org/w12-1-capstone/w12_1_capstone/src/50b6a6a6fa86d76d5fe4457d156f1419eefd171d/app/model/database/__test__/PatientRepo.test.js

[5] M. Patel et al. “Bitbucket: w_12_1_capstone/__test__”. Accessed: Sep 1, 2022. [Online]. Available: https://bitbucket.org/w12-1-capstone/w12_1_capstone/src/50b6a6a6fa86d76d5fe4457d156f1419eefd171d/app/model/database/__test__/

[6] S. Alpert. “React 16: A look inside an API-compatible rewrite of our frontend UI library”. Accessed Sep 3, 2022. [Online]. Available: https://engineering.fb.com/2017/09/26/web/react-16-a-look-inside-an-api-compatible-rewrite-of-our-frontend-ui-library/

Appendices

A1. User stories

These should be available at the Wiki with the full record of their development available to the marker. This section should provide the final version of them to make the report stand alone. In addition, provide the relevant url for the Wiki.

The user stories that were completed:

1. As a User, I want to be able to perform the balance test, so that I know if I have cerebellar damage [1]
2. As a User, I want to be able to perform a reaction test, so that I know if my reaction is bad [1]
3. As a User, I want to be able to perform the memory test so that I can know my immediate mental status [1]
4. As a User, I want to be able to access cognitive symbols, so that I am able to use the app easily and efficiently [1]
5. As a User, I want to be able to distinguish the colour in the app correctly, so that I won't do anything wrong due to my colour blindness [1]
6. As a User, I want to be able to see a menu of features offered, so that I am able to choose from them [1]
7. As a User, I want be able to perform assessment on my child, so that I can know if the child has a concussion from sports [1]
8. As a User, I want to be able to call 000, so that I can get help if the app shows that I have a concussion [1]

To see all our user stories, visit the wiki page [1].

A2. Unit testing summary

As mentioned in section 3, the program already has an extensive amount of unit testing. All unit tests may be found in the __test__ folder, and in order to run them you will need to run **npm run ci-test** in the terminal. This will also provide you with the code coverage.

A3. Acceptance Testing

1. UAT Scope (In Scope – Out of Scope)	
UAT - In Scope	**UAT - Out of Scope**
In Scope *List features that are tested* - Conducting all preliminary tests - Conducting daily Red flags & PCSS Checklist - Viewing Concussion Action Plan (with head bumps)	Out of Scope *List features that are not tested.* - Report (csv file) Generation - VOMS Tests

Table A3.1 User acceptance testing scope

2. UAT Assumptions and Constraints
UAT Assumptions
Assumption *List the UAT assumptions.* - **Test environment:** The test cases are conducted by 2 randomly selected users, and hard- and software is provided by the development team. - **Test documentation:** All UAT test cases are documented within this file. - **Error reporting:** Errors, failures and other flaws are reported to the monitoring team member directly.

Table A3.2 User acceptance testing assumptions and constraints

UAT Constraints

Constraint *List the UAT constraints.*
 	
- **Time frames**: Test results must be provided by Sep 3rd, 2022
 	
- **Resources:**
  o Human resources: 2 testers available for half hours.
  o Provided hardware: iPhone 13 Pro Max, XCode emulator.
  o Tested operating systems:
  ▪ IOS: 15.5, 15.6.1
  ▪ NPM: 8.18.0
  ▪ EXPO SDK: 44.0.0
  ▪ REACT: 17.0.1

Table A3.3 User acceptance testing constraints

3. UAT Risks			
Description	**Probability High|Med|Low**	**Impact High|Med|Low**	**Mitigation**
No CS/medical knowledge background	Low	Med	Simply follow the instructions on app.
Error handling: Test conductors are not aware on how to report bugs	Low	High	Talk to the team member directly.
UAT test failure	Low	High	Feature-complete development done before UAT test start.

Table A3.4 User acceptance testing risks

4. UAT Team Roles & Responsibilities		
Name	**Roles**	**Responsibilities**
Nancy Meng	Tester	Testing on iPhone
Gary Lou	Tester	Testing on emulator
Yupeng Zhang	Manager	Managing UAT Tests

Table A3.5 User acceptance testing team roles & responsibilities

5. UAT Entry Criteria	
ID	**Criteria**
5.1	- The development of the Preliminary tests, daily Reflags & PCSS Checklist, and the action plan features are completed.
5.2	Integration tests are completed
5.3	No high or medium defects are reported
5.4	All reported bugs should be fixed
5.5	UAT test environment (hardware, software) is ready
5.6	Testers are app users only purposed for testing
5.7	UAT test plan is available

Table A3.6 User acceptance testing entry criteria

6. UAT Requirements-Based Test Cases	
ID	**Test Cases**
6.1	Preliminary tests: • Open the app on iPhone • Select “Preliminary tests” from the menu bar • Follow the instructions on memory, reaction, and balance tests • Expected result: o This tester finishes all the tests o The results of all tests are presented to the user
6.2	Red flags & PCSS Checklist: • Open the app on iPhone • Click “Begin Check” button on the main page

	• Click “Ok” button on the pop-up alert window • Select symptoms from the list of checkboxes (randomly selected, since tester does not have a real concussion) • Click “Submit” button on the bottom of the page • Click “Save to new profile” on the result page after the checklists • Type in personal details and click “Submit” button • Click “View history” button on the home page to view the result • Expected result: o Personal profile successfully created o All selected symptoms are saved, viewable in profile and following instructions for the use are prompted
6.3	Concussion Action Plan • Open the app on iPhone emulator • Select “Concussion Action Plan” from the menu bar • Click “Check Symptoms” button on the pop-up alert window • Have a read through the head bumps information on the page • Click “Back” button on the bottom of the page • Select “Concussion Action Plan” from the menu bar • Click “View action plan” button on the pop-up alert window • Have a read through the information of each zone on the page • Expected result: o The alert window successfully popped up o All the info on the Head bumps and action plan pages are clear and viewable

Table A3.7 User acceptance testing requirements

7. UAT Test Results				
ID	**Test Cases**	**Pass/Fail**	**Tested By**	**Date Tested**
7.1	Conducting of Preliminary tests	Pass	Nancy Meng	03/09/2022
7.2	Conducting of Red flag & PCSS checklists	Pass	Gary Lou	03/09/2022
7.3	Viewing of Concussion Action Plan	Pass	Gary Lou	03/09/2022

Table A3.8 User acceptance testing test results

A4. Usability Testing

1. Usability Scope	
In Scope	**Out of Scope**
In Scope • Give the participants tasks related to the application features/navigation, such as how to access preliminary tests, how to create a profile etc	Out of Scope • VOMS Tests

Table A4.1 Usability Scope

2. Participants			
Participant ID	**Name**	**Age**	**Prior knowledge of concussions**
1	Coco Banks	25	Has extensive knowledge of concussions
2	Monia Loenn	24	Has little knowledge of concussions

Table A4.2 Usability participants

3. Usability Tasks			
ID	**Testing**	**Observations (ID: 1)**	**Observations (ID: 2)**
1	If you wanted to create a profile, where would you go?	From the home page, the user pressed the menu icon in the top left corner followed by “Login”. The participant then entered her details and clicked on “Submit” which gave an error message. The participant then clicked on “Create Login”, and filled in details again and pressed “Submit”.	From the home page, the user pressed the menu icon in the top left corner followed by “Login”. The participant then clicked on “Create Login”, filled in her details and pressed “Submit”
2	Imagine if you felt nauseous and wanted to check if you have a concussion. How would you do this?	The participant clicked on “Preliminary tests” from the side menu, and performed all of the tests.	The participant clicked on “Preliminary tests” from the side menu, however she did mention that it was difficult to know if she was meant to press “VOMS Tests” or “Preliminary tests” due to that she does not have a lot of knowledge about concussions, hence she did not know what VOMS tests are.
3	For each preliminary test (memory, reaction and balance), how would you start the tests?	For the memory tests, the participant pressed the white “Start button”. For the reaction tests, the participant pressed the red circle. For the balance tests, the participant pressed the red circle.	For the memory tests, the participant pressed the white “Start button”. For the reaction tests, the participant pressed the red circle. For the balance tests, the participant pressed the red circle.
4	If you wanted to have a look at the Concussion Action Plan, how would you do this?	The participant clicked on “Concussion Action Plan” from the side menu, and was able to successfully access the action plan	The participant clicked on “Concussion Action Plan” from the side menu, and was able to successfully access the action plan

Table A4.3 Usability Tasks

4. Summary of Observations	
ID	**Observation**
1	It seems like the participants may have been confused by how to create a profile. For example after clicking on “Login”, the “Submit” and “Create login” button seem fairly similar and it is not clear whether or not details need to be filled in before pressing “Create login”. Participant 1 had to fill in details twice because there were confusions around this, so a suggestion would be to change this part of the application to make it more clear how to login and create a new profile.
2	One of the participants mentioned that it was difficult to know the difference between preliminary tests and VOMS tests due to that she does not have any prior knowledge of any of this. A suggestion would be to include some more information about VOMS tests, and the difference between preliminary and VOMS.
3	It seems like both participants were able to successfully start the preliminary, which indicates that this part of the application has a high usability and is easy to understand
4	As for the previous task, it seems like the participants did not have any issue finding the concussion action plan. This indicates that it is accessible within the app and relatively easy to find.

Table A4.4 Usability Summary of Observations

A5. System Structure Overview Diagrams

A5.1. Previous System Architecture Design

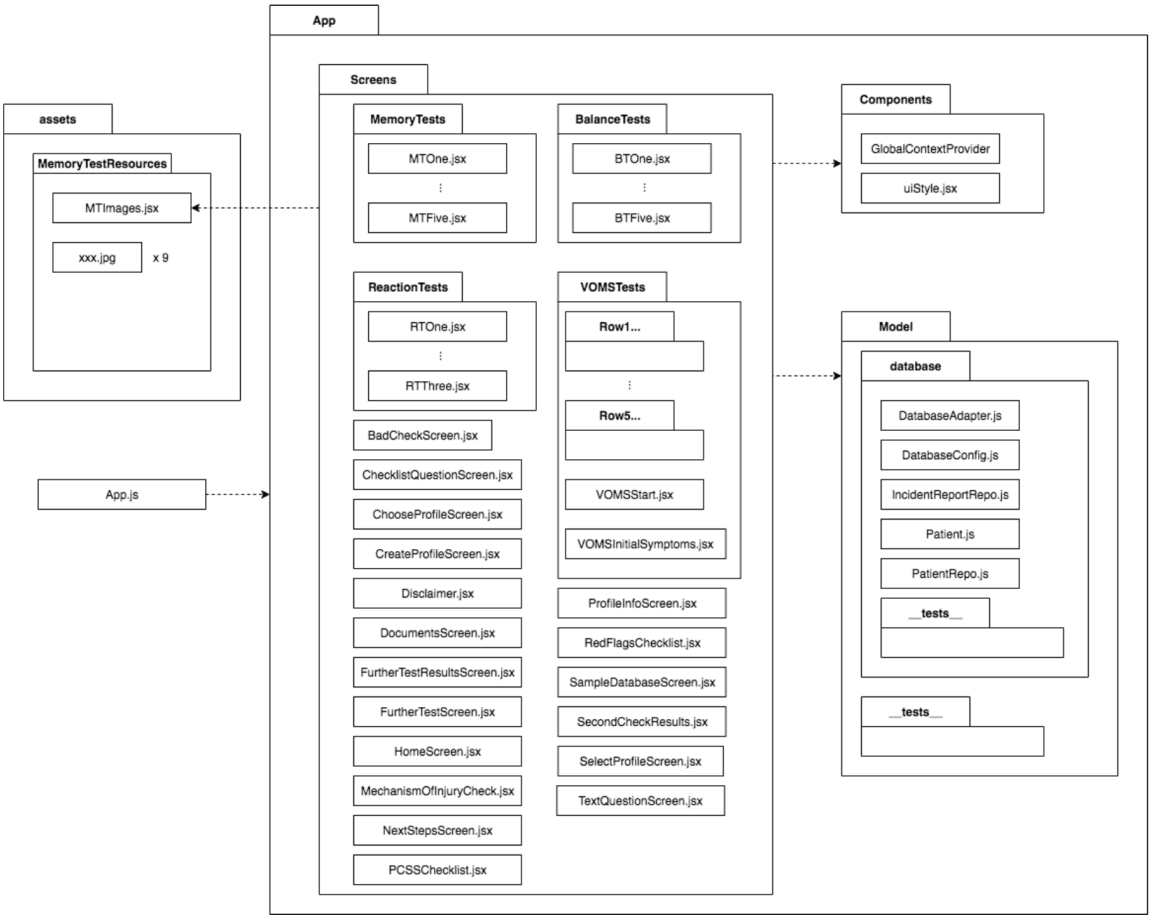

Figure A5.1 Previous System Architecture Design

A5.2. High-level UML diagram for current system structure

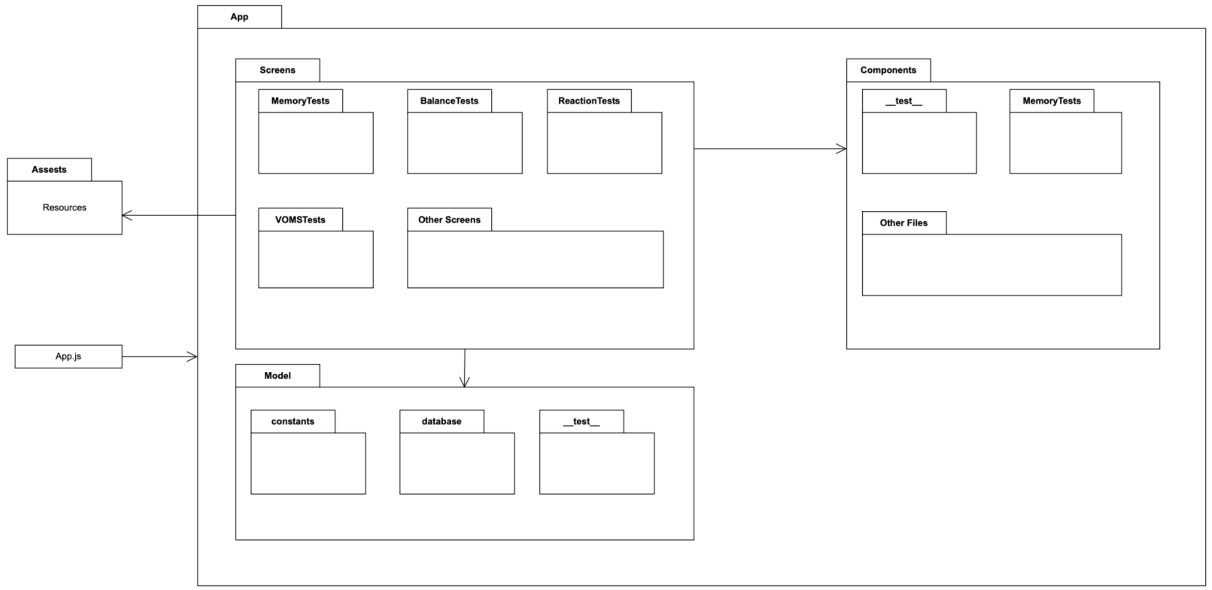

Figure A5.2 UML Diagram

A5.3. Detailed UML diagram for current system structure

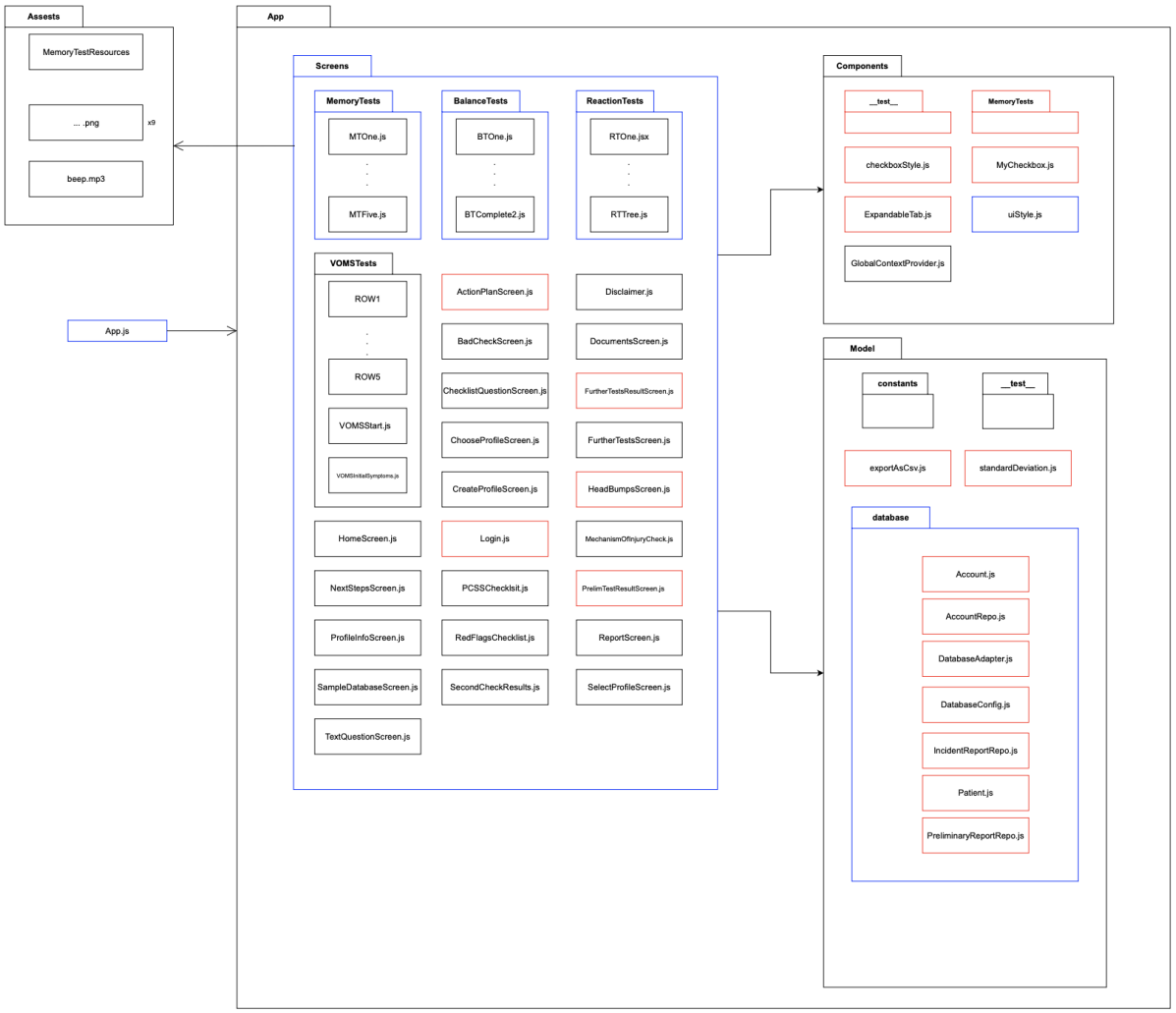

Figure A5.3 Detailed UML Diagram