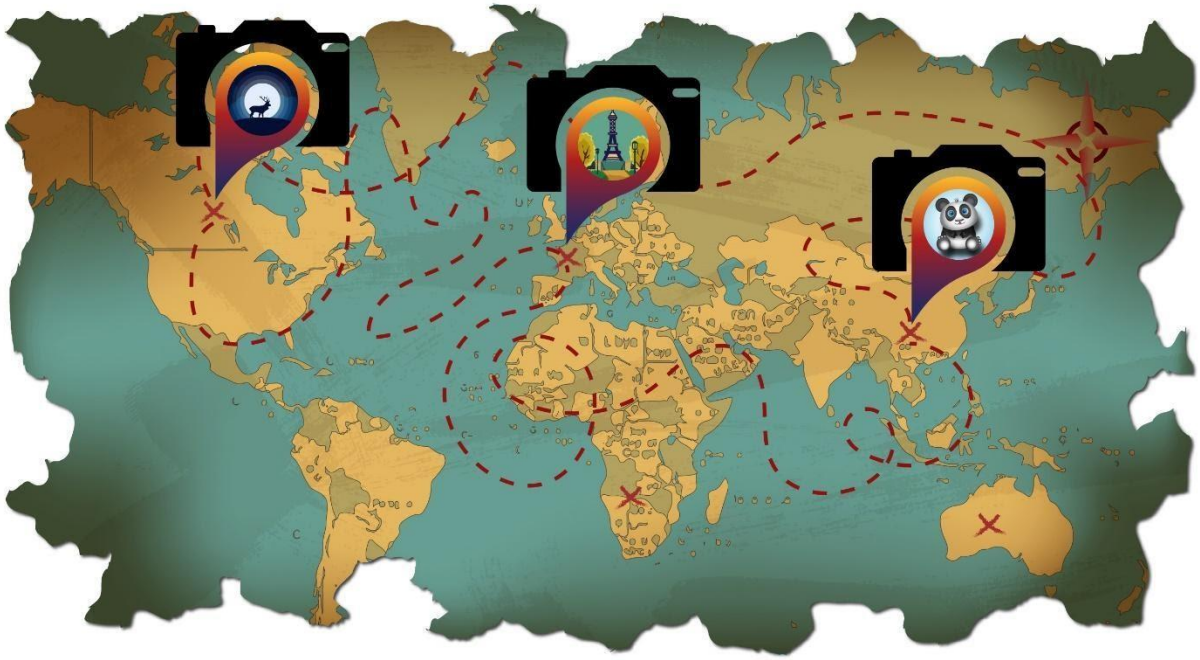


PixPlace
Social Hub

Stage 2 (Halfway) Report

Company Site

https://aaronmolesbury.github.io/PixPlace-Company-Website/website/home.html

Group 3 | PixPlace Software Development

Ghazi Yusaf, Aaron Molesbury, Magnus Mackay,

Tania Henderson, Qianqian Fan, Jose Carlos Fernandes

Project Coordinator: Ron Petrick | **Group Manager**: Jessica Chen-Burger

Table of Contents

1. Introduction

1.1 Purpose

This document is the progress report for the social media platform "PixPlace", describing the status of the application as of the project's stage 2 deadline by detailing what functionalities have been implemented in the app from the design stage and the development team's plan for the next stage. This document will include a front-end and back-end overview of the application's design and technologies and will feature various diagrams and in-depth descriptions of the app's purpose and performance.

1.2 Audience

This document's intended readers are:

1. PixPlace development team - to be used as a reference for the project's progress and design.
2. The client and their company - to inspect the current and future design of the application.
3. Project Manager - to oversee that appropriate progress is being made by the group.

1.3 Scope

The team will create a competitive image-sharing social media hub that developed for the latest internet browsers and mobile operating systems. As of stage 2, the team has developed the main pages of the app to a functional level, including a login page, a home page, a leaderboards page and a challenges page. The app also has a search function that allows the user to search for users, channels and photos that have been tagged with the searched keyword(s).

1.4 Overview

This document illustrates the details of the platform design and functionality that the PixPlace Software Development team devised in stage 1 and how we implemented them in stage 2, starting with an overview of the design detailing the different software that was used for the application and why we used them.

Next is the overview of the front-end of the application which briefly describes each of the displayed screens in the app and their current development status. Following this is the overview of the back-end, beginning with an explanation as to why the team made a change in solution from Kubernetes to Firebase as well as providing details of the current functionalities of the back-end.

The document then goes on to describe the testing that was performed for the application including an integration test using Flutter's packages and... *Note: give more details here if/when more tests are created.*

1.5 Preliminary Notes

Abbreviation and definitions

iOS	Operating system for all mobile Apple devices
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
SCRUM	Form of Agile development that relies on short deadlines of iterative work, known as "Sprints"
JSON	JavaScript Object Notation
PostgreSQL	Relational database management system conforming to SQL compliance
SDK	Software Development Kit
Windows, Macintosh, Linux	References the three most common desktop computer operating systems.
API	Application Programming Interface
Git	Widely used version control system.
MVC	Model View Controller
MVP	Model View Presenter
F	Functional
NF	Non-Functional
R	Requirements
UR	User Requirement
SR	System Requirement

Priorities

🔴	**High Priority**
🟡	**Medium Priority**
🟢	**Low Priority**

Progress

✓	**Completed**

~	**In Progress**
x	**Not Started**

1.6 References

Team 3 (2020). The Bid Document.

2. Design Overview

2.1 General Overview

The development stage is now halfway and the consensus within the group is that the development process has accelerated over the past few weeks. The project's plan was slightly rolled back due to resource changes, but the recent changes are expected to comply with all the requirements specified without any significant exception.

The group has been developing the app using Android Studio installed with flutter and its plugins. The Android SDK, along with browser debugging has been used to test the application. Visual Studio Code has also been used for developing by some of our group members.

The mobile application is where most of the resources have been invested in because it is the project's main deliverable. The development of the application has been very productive so far, as many main interfaces were already set up with its basic functionality. Some of the content is still hard-coded but the connection to the backend is currently being established.

The Flutter framework has proved very robust, resourceful, and yet very flexible and expressive. Its greatest advantage is its intuitive and declarative API which eases the programming experience and increases code reusability. The Flutter community is highly supportive by providing examples of many use cases and by producing and distributing useful packages for additional functionality.

The back-end is also being developed simultaneously to close the first development iteration as soon as possible. This is particularly important since the agile method is successful only if the process is based on rapid development cycles. The back-end services are already being deployed as any user can now register or login into the application.

The Firebase platform is the infrastructure supporting all the services to this application, instead of what was originally planned. The decision was made to roll back the Kubernetes docker container cluster in favour of a simpler stack due to unexpected events related to the team. After some research, assessment and negotiation, Firebase was the solution chosen as it provides a native interface to Flutter. This platform has an extensive set of tools which speed up development but also ensures the application's security and scalability. These are vital attributes necessary for a highly available image-sharing social network that manages personal data of its users.

Currently these are the minimum dependencies for the system administrator to successfully manage the PixPlace system:

- Firebase account, access to Firebase console
- Flutter 1.17.0
- Dart SDK 2.8

These dependencies are currently all that is required to run and deploy the system. Considering the malleable nature of the project, these dependencies may be updated in stage 3 as more functionality is added.

The dependencies for the user are dependent on upon the platform they wish to run the application on. Currently the application is not available to be run as a standalone app for Android and iOS, a web application or as a standalone desktop application for Windows, Macintosh and Linux. The application will not run on Windows mobile OS as this is not supported by Flutter. Considering that Windows phone users consist of less than 1% of the phone market, the team has decided that developing for the platform would not be worth the extra cost and time. The dependencies for each platform are as follows:

- Android 5.0 (Lollipop), API level 21 or higher
- iOS 7.0 or higher
- Windows 8 or higher
- Linux, most modern distros supported
- macOS 10.14.4 or higher

Any modern web browser with JavaScript will be able to run PixPlace. Notable browsers with tested support include:

- Google Chrome ES5.1 or higher
- Edge 45.12 or higher
- Safari
- Safari 5.1 or higher

https://gs.statcounter.com/os-market-share/mobile/worldwide - this link shows that the windows mobile share of devices is miniscule, therefore there is no point developing for it.

2.2 Functional requirements breakdown

2.2.1

Progress	F-R1	Registration
✓	F-UR1-1.1	🔴 **Register user account** Anyone who uses the app will be able to create a user account using an email and password.
✓	F-UR1-1.2	🔴 **Create username** Users will be able to assign themselves a username, as long as the current username has not been registered.
×	F-UR1-1.3	🟡 **Create user biography** Users can create 150-character long biographies to describe their account.
×	F-UR1-1.4	🟢 **Create user avatar** Users will be able to assign themselves a custom avatar that will act as their profile photo, from a selection of available avatars.
✓	F-UR1-2	🟢 **Verify email** New users could be sent a one-time verification email to confirm their email address, after which they would be allowed access to the platform
✓	F-UR1-3	🔴 **Login and logout** Users will be able to login into their account using their username or email along with their password and can logout at any time.
~	F-UR1-4	🔴 **Age restriction** Users must be at least 13 years old in order to create an account and use the platform, registration will confirm user age through their date of birth.
~	F-UR1-5	🔴 **One account per email** An account on the platform can only be linked to one email, attempting to create a new account with an already registered email will not be permitted.
~	F-UR1-6	🔴 **Delete account** A user account can be deleted at any time, once deleted all user information will be removed from servers, and the email linked to the account shall be available again. Account deletion is irreversible, users will be warned before proceeding.

F-UR1-1.1, F-UR1-1.2

Users can create and register an account on the system through a login page. The account will be created through our back-end system, Firebase Authorization. In addition to the standard login procedure the user will also be able to reset their password via their email, should they forget it. Usernames are not unique; this was decided to give users more freedom of choice.

F-UR1-2

Users will be sent an email verification via Firebase. They will not be able to access the platform without following the link in the email to verify their account. Should the email not be sent, the user will be given the option to resend their email verification.

F-UR1-3

Users can now only login into their account via their email. This change is necessary because usernames are no longer unique.

F-UR1-4

Age restriction will likely be implemented through the system's terms and conditions policies. The sign-up screen requires users to consent to the terms and conditions before creating an account. These terms are in the process of being written and cannot currently be viewed by potential users.

F-UR1-6

System administrators can delete any account on the system. Currently users cannot delete accounts they have created themselves.

2.2.2

Progress	F-R2	Platform use
×	F-UR2-1.1	🔴 **Post images** Users registered to an account with the system shall be able to post photos.
~	F-UR2-1.2	🟢 **In app camera** Photos can be taken and then posted directly through the app if the user grants the app permission to access the devices camera.
×	F-UR2-2	🔴 **Channels** Users can create channels, which they will be automatically granted admin rights for. Photos within a channel can be arranged however a user like. Other users can be invited to the channel and may post and interact with others in the channel. Only users within the channel will be able to see or interact with the channels content.
×	F-UR2-3	🟡 **Post visibility** Users can set the visibility of their posts to one of four levels: ▪ **Global:** Anyone will be able to view the picture. ▪ **Local:** Only users within a certain distance from the poster will be able to see the post. ▪ **Friends:** Only the users' friends will be able to see the post. ▪ **Private:** Only the user will be able to see the post. Posts with a visibility of friends or private will not be eligible for challenges and cannot earn the user experience.
×	F-UR2-4	🔴 **Post location** Users can attach the location where the photo was taken to the post. Posts that do not have an associated location will not be eligible for certain challenges.

F-UR2-1.2

Users on Mobile platforms can use their built-in camera to take pictures directly within the app. This functionality does not currently work on browser or desktop.

2.2.3

Progress	F-R3	Navigation
~	F-UR3-1	◉ **Homepage** The platform will create a personalised user homepage, that will recommend other channels and image posts from users based on their previous use of the platform.
~	F-UR3-2	◉ **Search** The platform will have a search option that will allow search for users, channels or photos with that tag. When searching by tag, the platform will also allow for displaying results from related tags.
×	F-UR3-3	◉ **Friends** Users can navigate to a friend's page by either looking through their friends list, or by searching their friends list for a specific friend.
~	F-UR3-3	◉ **Profile** Contains all posts from the user, and all rewards they have achieved.
~	F-UR3-4	◉ **Bottom anchored navigation bar** All the different subsections of the app will be directly accessible through a navigation bar always anchored to the bottom of the app.
~	F-UR3-5	◉ **Top anchored navigation bar** Access to user settings and notifications will be found on a navigation bar anchored to the top of the app.

All visual parts of the app have been created and tested with static data. Basic functionality across the all parts, but nothing has been tied to the back-end. The focus of the last stage will be taking these designs and connecting them with our Firebase back-end.

2.2.4

Progress	F-R4	Interactions
~	F-UR4-1	🔴 **Comments** Users will be able to comment on pictures other users have posted.
×	F-UR4-2	🔴 **Point-based rewards** Users will be able to award points to pictures other users have posted with points.
×	F-UR4-3	🟡 **Sharing** Other platform users should be able to share other users' photos to their channels.
×	F-UR4-4	🟢 **Saving** Users can save posts from other users or their own posts, with a section of the platform dedicated to viewing all previously saved posts.
×	F-UR4-5	🟢 **Tagging** Posts will be tagged based on what animal/object they contain.
×	F-UR4-6	🟢 **Reporting** Users can report other posts that break the policies and conduct of the platform. Moderators will be notified of reported posts.

F-UR4-1

Static comments are available in the current implementation of the feed page, not currently functional with the backend.

2.2.5

Progress	F-R5	Game
~	F-UR5-1	◉ **Challenges** Users can complete daily, weekly and monthly challenges to earn experience. Longer timeframe challenges will be more difficult but will offer more experience as a reward.
~	F-UR5-2	◉ **Experience** Experience is the main reward for the game side of the app. It is used as a gauge of success and loyalty, and as experience is gained the user will level up and earn new rewards and benefits.
~	F-UR5-3	◉ **Level system** Users can gain levels by earning experience, each level requires more experience than the last to reach. The maximum achievable level is 100.
×	F-UR5-4	◉ **Rewards** Users can earn rewards for reaching certain milestones within the game (e.g. completing a certain number of challenges, reaching a certain level). Rewards can be placed on a user's profile for others to see, and take the form of avatar cosmetics, or badges/medals/trophies.
~	F-UR5-5	◉ **Leader boards** Multiple leader boards will be available to allow users to compare their progress to another subset of users. Such leader boards include: ▪ **Global:** Compare against all other users on the platform. ▪ **Local:** Compare to all other users within a certain vicinity (country, town, area). ▪ **Friends:** Compare to all the users on their friends list. All leader boards can be sorted by either score, pictures uploaded, or rewards earnt.

Challenges page has been implemented; actual challenges available on the system are still in the design process. A set of challenges for each time frame will be developed in stage 3. The rest of the requirements have dedicated visual components but currently no functionality.

2.2.6

Progress	F-R8	Analytics Data
~	F-SR8-1	◉ **Usage data** Details of user activity (time of use, number of images posted per day, concurrent users, number of registered users etc.) will be collated into an analytics report available to administrators.
~	F-SR8-2	◉ **Report actions** Administrators will be able to provide feedback based on analytics reports, and the platform will be changed accordingly in an effort to better suit the needs of the client/users.

System administrators have usage and report data provided to them via Firebase. Data is limited to what Firebase is able to provide, but in stage 3 a custom API could be placed on top of Firebase to provide more detailed analytics. Firebase also offers more detailed analytics through a paid subscription that is not currently being used.

2.3 Non-Functional Requirements Breakdown

2.3.1

Progress	NF-R1	Hardware
✓	NF-UR1-1	◉ **Cross-platform support** The system should operate on the following platforms: **Mobile Devices:** ▪ Android ▪ iOS ▪ Windows While also providing a high standard of user experience across each platform.
✓	NF-SR1-2	◉ **Standard hardware specifications** The standard operating environment for the system should include the following hardware: ▪ Camera ▪ GPS sensor ▪ Wireless network interface controller
✓	NF-SR1-3	◉ **Storage** The system capacity will be scalable to account for increasing numbers of users and data.

NF-UR1-1

Flutter has provided a platform that allows for cross-platform support, though the app will not function on Windows phones. This was a team decision as Windows phones only account for an extremely small percentage of mobile users.

System has been tested and is functional on devices that meet these requirements as of now.

Firebase offers a dynamic database that can accommodate many users. Free Firebase accounts are limited, but in future a paid subscription to Firebase would allow for a larger database.

2.3.2

Progress	NF-R2	Software
✓	NF-UR2-1	◉ **Software support** The system should operate on the following software: **Web browsers:** ▪ Google Chrome ▪ Mozilla Firefox ▪ Apple Safari ▪ Microsoft Edge While also providing a high standard of user experience for each software.
✓	NF-SR2-2	◉ **Firmware support** The system should support firmware releases from the last 3 years for each platform the app is available on.
✓	NF-SR2-3	◉ **Model-View-Controller** The system will utilise MVC design as a framework to breakdown the system into related components.

NF-SR2-2

Flutter works with versions of all required platforms from the last three years.

MVC has been used to break down the system during the development process, to allow team members to focus on their most comfortable aspects of design.

2.3.3

Progress	NF-R3	Data
~	NF-SR3-1	◉ **Data storage** User data will be stored securely on a remote server.
×	NF-SR3-2	◉ **Backup of data** All critical user and platform data will be securely backed up to another location, in case of emergency.
~	NF-SR3-3	◉ **User details** User details will not be directly stored, session data will be used to create a seamless experience when returning to the platform on a previously used device. Any user logins from a new device will be alerted to the user through their email.

Data storage is in progress using Firebase. Currently no backup exists of the database, and this will be implemented in stage 3.

2.3.4

Progress	NF-R4	Security
~	NF-SR4-1	◉ **Secure platform** The platform will use encryption and web application firewalls to prevent data breaches from malicious attackers.
~	NF-SR4-2	◉ **Secure login** Access to the platform will be through secure protocols (such as HTTPS) to protect sensitive user data.
×	NF-UR4-2	◉ **User visibility** Users can choose the visibility level (Global, Friends or Private) of their account information.
✓	NF-UR4-2	◉ **Password guideline** Users must create passwords in accordance with the following criteria: ▪ **Length:** Password must contain at least 8 characters. ▪ **Numbers:** Password must contain at least one number. ▪ **Special characters:** Password must contain at least one special character (!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~). ▪ **Case:** Password must contain at least one uppercase and one lowercase letter. In order to prevent others from easily gaining access to their account:

Firebase's built in security will be used to keep the platform safe. More precautions can be added on top, to allow the system to be as secure as possible.

The password guideline has been slightly changed; all guidelines have been met besides the use of a special character. The passwords seemed to be secure enough without meeting this requirement. This could be changed in stage 3 if it poses a significant risk of account breaching.

3. Front-end Overview

The application design is modern and intends to appeal to young people that are passionate, and image driven. Based on this, we are working on producing a design that has an appealing and sleek interface. Additionally, this interface will be consistent across Android, iOS and web, thanks to the front-end framework.

The team has worked intensively in the mobile application and has already produced layouts for every interface. Most of the interfaces do not have the full MVC implemented yet but it is as planned at this stage. The team is currently connecting the back-end to the front-end interfaces.

3.1 Splash screen

The splash screen is a particularly useful screen because it shows the application's logo while the application loads, enhancing the look and feel. This improves the users' experience by providing a level of responsiveness and makes the application more visible for people around, creating a strong brand identity.

3.2 Login / Register Page

This page is close to the final release version because it has all the functionality implemented, both its local MVC and the Firebase registration and authentication services. It has a very sophisticated interface which the team believes that will engage users right away.

3.3 Camera Page

The camera is the functionality that brings everything else in the application together. We have integrated the native camera with the application, and it is now possible to view its output in real time onto the page's screen. The current implementation is still basic although major work is planned to start right after the deadline.

3.4 Profile Page

This page already has the main layout set up with the avatar, a circle progress bar, a button menu and area for the user's bio. It is still a work in progress because its MVC is not implemented but it has the main functionalities required at this stage. This page might have more additional features in the next stage of development such as integration of the user's posts, collections and challenges.

3.5 Feed Page

The development team expects the users to spend most of their time in the feed and so there has been a lot of work done here to make it fit their needs and expectations. This interface is composed of individual posts that can be liked, commented, shared or even saved in their own collections. The layout is now finished but there is still development yet to be carried out in its local MVC and Firebase database.

3.6 Challenges Page

This page now has 3 main tabs that divide the daily, weekly and monthly challenges. Each tab can then list all its challenges for users to get assigned and track their progress. The content being rendered is still hard coded, but it will eventually get data from a real time database during the next stage.

3.7 Leaderboards Page

In this page, the positions of users can be displayed into two different tabs: Global and Friends. It can list many users with their respective experience points, although it only displays hard coded content. The next step will be to retrieve the user's points and sort them into each table.

3.8 Settings Page

The settings page has now the main functionalities needed for a basic use case. It includes general options about location, notifications, account and it informs users about the privacy policy and help page. This page's MVC will be implemented in the next stage.

4. Back-end Overview

The back-end planned during the first stage was to deploy a Kubernetes cluster running multiple docker containers as this was the best solution for development but also for deployment, however this solution was dropped due to issues with the team. It was negotiated that this infrastructure should be rolled back to be replaced by a more accessible solution to everybody.

The solution chosen was Firebase because of its easy integration with Flutter. After installing the Firebase native Flutter package, the mobile application now has access to any service provided by this infrastructure. Although the developing team is still exploring the infrastructure, there are already features that are or will be integrated into the application.

4.1 Registration / Authentication

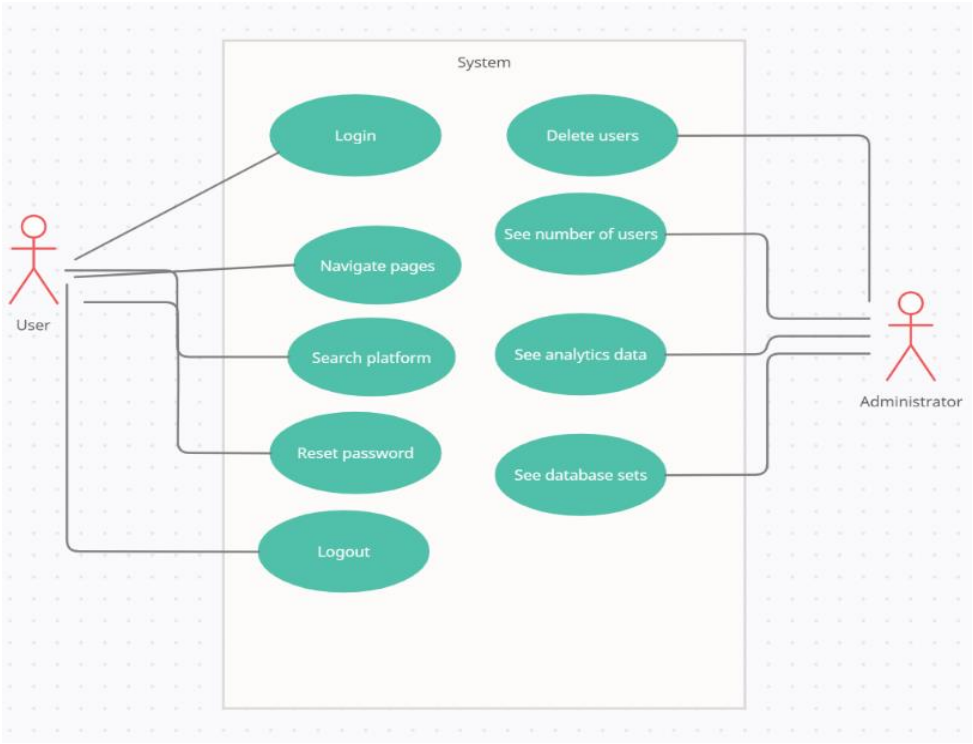
The app now allows users to sign-up and authenticates their details before they can engage with the rest of the application. This is provided by the Firebase SDK, which provides packages for integrations for web and mobile applications. In order to use Firebase in Flutter, it is required to use the official Firebase package. This package connects the mobile application directly with the Firebase platform to track the user's registrations and logins and send confirmation emails.

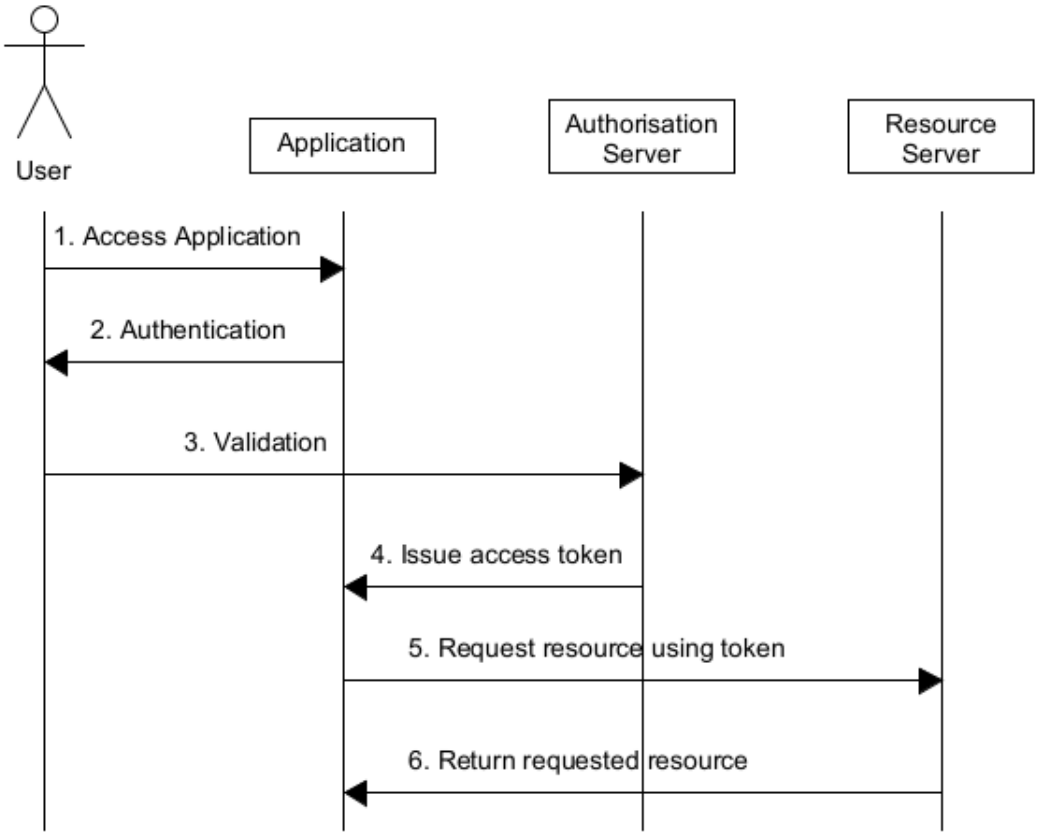
4.2 Cloud Storage

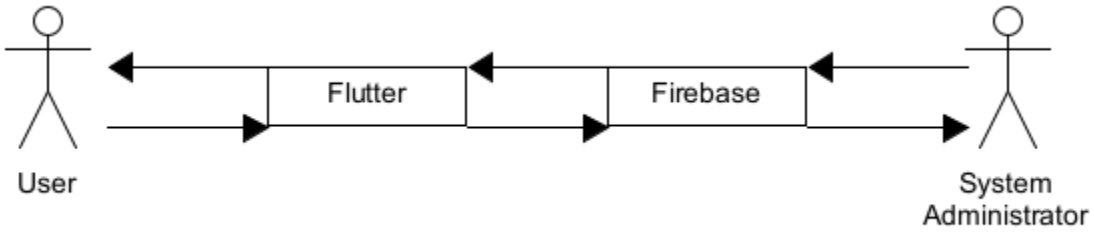
The mobile application is expected to deal with a large dataset of media, especially images. This would be handled by some sort of cloud-based hosting, but this decision made the hosting decision easier since the Firebase platform has a cloud storage service with easy integration.

4.3 Diagrams

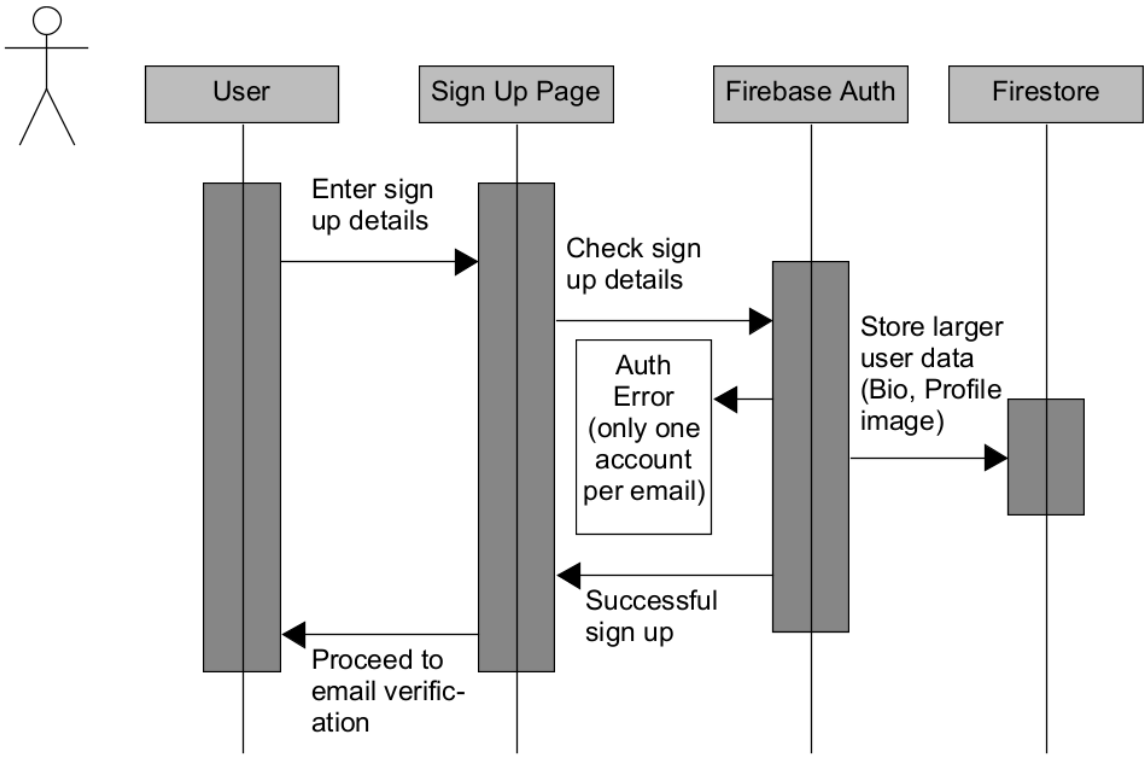
Use Case diagram Showing the Current Functionality Achieved


System
Login
Delete users
See number of users
Navigate pages
Search platform
See analytics data
Reset password
See database sets
Logout
User
Administrator

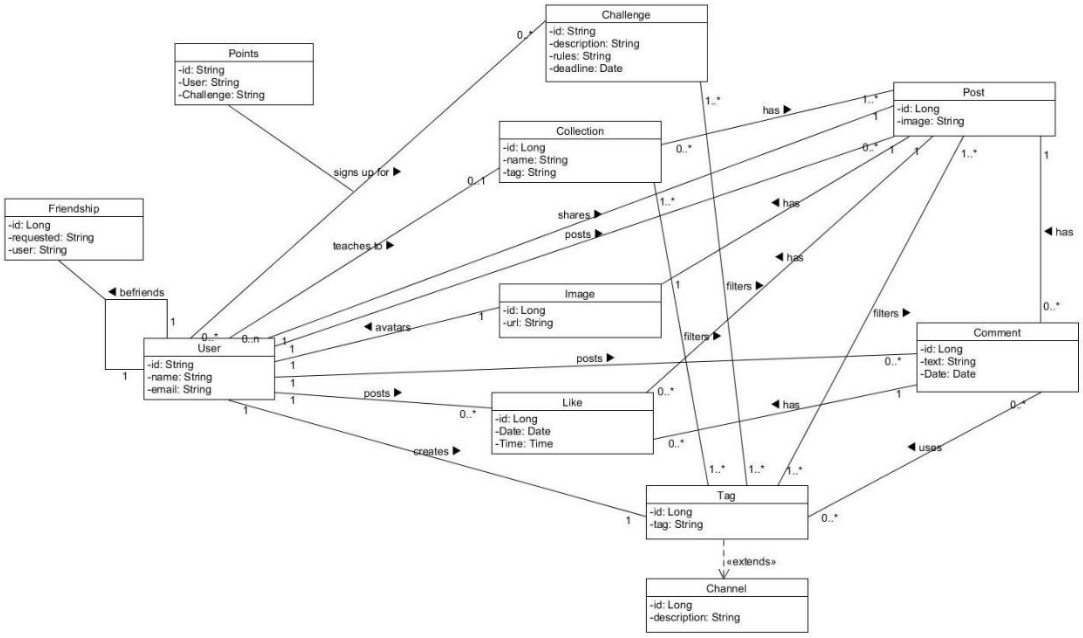

User
Application
Authorisation Server
Resource Server
1. Access Application
2. Authentication
3. Validation
4. Issue access token
5. Request resource using token
6. Return requested resource



Sequence Diagram for Login



Entity Relationship Diagram



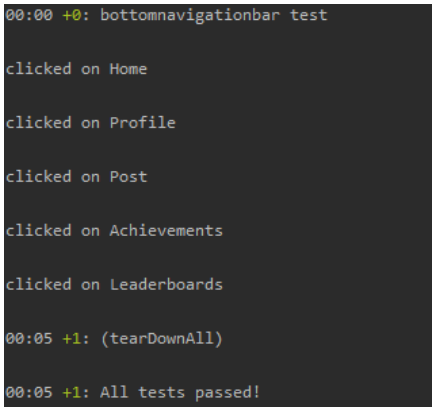
The text below shows the app structure for the PixPlace. There has two main blocks in the architecture: users interface and back-end that all of them can be subdivided. Users interface can be devided into 4 parts to illustrate as what users can do with PixPlace, like Camera Page, Challenge Page, Leaderboards Page and MVC part(will be completed and realized totally in stage 3). The MVC part includes Login Page, Profile Page, Feed Page and Setting Page that all of these functions use MVC to work with in. In the block of back-end which is the important part to make every function come true can be split into 2 parts as Registration to save their sign-up and sign-in data, and Cloud Storage to deal with uploaded images mainly.

5. Testing

At this stage, much of the testing was conducted manually by the team as most of the features that could be tested with unit, widget and integration testing either were not fully implemented until closer to the deadline or will not be implemented until stage 3.

5.1 Bottom Anchored Navigation Bar

For testing the bottom anchored navigation bar we used the flutter_driver package to perform an integration test that verified the functionality of the navigation buttons. This entailed using the Flutter driver to tap on each of the buttons to switch pages and print a message when it was successful in doing so each time. Below is an example of a successful test's output:



5.2 Login / Registration

To test the login/registration page we input both valid and invalid details to verify that the application would accept the valid details and reject the invalid details (and by extension, inform the user that the details were rejected). Using Firebase, we restricted what kind of details would be accepted, for example: usernames would not be able to have an underscore "_" at the beginning or the end, only suitable email formats would be accepted, and passwords would need to be at least eight characters long and require at least one number, one uppercase and one lowercase character. We also tested that the application would

let the potential user continue to register if they ticked the box that acknowledged they had read the terms and conditions.

These tests were successful, showing that the application would accept valid details and create the account for the user, while it would not accept invalid details and would not create the account if the terms and conditions box was not ticked. The potential user would also be informed of the rejection with a message in red text appearing above the appropriate text fields where they had input their details.

Another aspect of the registration process that was tested for was duplicate account details. While the team decided that usernames would no longer have to be unique, if a potential user attempts to sign up to PixPlace with an email address that was tied to another account a pop-up would appear on the screen that informed them that the email address was currently already in use.

The final aspects of the login/registration page that was tested for was the email verification and password reset. Upon successfully registering for an account, a verification email would be sent to the user's email address that included a link that, once clicked on, would verify the user and open their account. For the password reset, should the user end up forgetting their password there is a link on the login page that opens a page in the user's browser where they can follow the instructions to receive an email with another link that would allow them to enter a new password for their account. The team tested these aspects and were able to confirm that the verification email and the process to reset a user's password is successful (although those emails may accidentally be sent to the user's junk mail, depending on the user's mailbox).

5.3 Emulator Conditions

The team also tested the application's capabilities when the Android Emulator's conditions were manipulated. The following conditions were simulated:

- Different battery levels (100%, 75%, 50%, 25%, 5%)
- Receiving a call
- Receiving an SMS

While interacting with the application as a normal user would, the team did not notice any difference in performance quality under these conditions. Additionally, if the incoming call was accepted and then cancelled, the app successfully returned to the state it was in prior to the call.

5.4 UI Consistency

As development of the application progressed, the team was continually testing the consistency of the UI, such as the layout of the navigation bars and icons, the keyboard size, the font of the text, et cetera. At this stage, the team is satisfied that there are no interferences with the intended design of the UI on the main pages of the app.

6. Future Milestones

6.1 Stage 3

The development team have estimated that approximately 50% of PixPlace's functionality has been successfully achieved, including the application's login/registration and main pages that the users will interact with. To continue with the development in stage 3, the team will dedicate their time towards implementing the rest of the functional and non-functional requirements specified in The Bid document. The team believes that once the database has been fully implemented, that will drastically increase the functionality of the application as it is the database that will allow most of the application's main features to function. In addition, the team will work towards further developing the security of the app in order to ensure the guidelines are being followed, for example the photo validation that checks for no visible human faces (OpenCV).

Another feature of the application the team will focus their time on in stage 3 is the game-like aspect that will use the experience points the users can collect when they interact with the app. As described in the team's The Bid document, the game will involve the collection of experience points (abbreviated to "XP") that can be gained whenever a user completes a challenge - Daily, Weekly or Monthly with the amount of XP that can be earned increasing with the time frame. This will also use the software OpenCV.

7. Appendices

7.1. Roles and Responsibilities

Name	Official Role	Responsibilities
Ghazi Yusaf	Organisational Manager	Ensure that tasks are allocated, and that progress is being made.
Jose Carlos Fernandes	Technical Manager	Make key technical decisions for the application.
Aaron Molesbury	Liaison	Contact members of staff and keep group in touch.
Tania Henderson	Reporter	Document progress of the application, take minutes during meetings.
Magnus Mackay	Game developer	Designer of the game-like aspects.
Qianqian Fan	Testing	Testing the application's functions.

7.2. Work Done

Name	Work Done (Stage 2)
Ghazi Yusaf	Organised meetings, created tasks in ClickUp and managed sprints, instructed all the other members through installing Flutter and Android Studio, managed the GitHub repo, merged other members work together from their branches, created entirety of the login and sign-up page including front and back end, connected login page with Firebase to allow for a fully functional user creation system. Created most the diagrams, added the requirements breakdown and wrote the deployment section in the Progress Report. Wrote the abbreviations and changed system to MVP.

Jose Carlos Fernandes	Front-end developer in stage 2, worked on main pages and Settings page of application. Created ER diagram and worked on overview sections of the Progress Report.
Aaron Molesbury	Front-end developer in stage 2, created challenges and feed. Helped develop other pages. created company website, helped to write the Progress Report.
Tania Henderson	Helped to write the Progress Report, assisted with the website, tester for the application in stage 2.
Magnus Mackay	Front-end developer in stage 2, contributed search.
Qianqian Fan	Minor changes to report.

8. Changelog

8.1. Revisions

Date	Author(s)	Changes
22/01/21	Tania Henderson	Document created.
23/01/21	Tania Henderson	Introduction and Audience written.
24/01/21	Tania Henderson	Added a link reference and headings.
29/01/21	Jose Fernandes	Design Overview section created and finished. Front-end overview section started.

30/01/21	Jose Fernandes	Working on front-end overview.
31/01/21	Jose Fernandes	Finished front-end overview. Created and finished back-end overview.
31/01/21	Tania Henderson	Changed document format.
01/02/21	Tania Henderson	Overview. Started writing the Testing section.
03/02/21	Ghazi Yusaf/Aaron Molesbury	Refactored and neatened report and added FR and NFR analyses.
03/02/21	Tania Henderson	Added to Future Milestones. Finished Testing. Added page numbers and ER diagram.
04/02/21	Ghazi Yusaf	Added requirements breakdown, deployment section and most of the diagrams.
03/02/21	Jose Fernandes	Created and developed entity relationship diagram that matches the system's design
04/02/21	Tania Henderson	Added tables to Appendices.
04/02/21	Jose Fernandes	Changed entity relationship diagram