



StudentPeak
CO600 - Technical Report

Tyrone Friday (tf289), Nicole Akanbi (na499), Israel Adekunle (ia271)
and Ayodeji Fasuba (aof5)

Supervisor - Kemi Ademoye (K.Ademoye@kent.ac.uk)

Contents Page

Contents Page.....	2
1. Abstract.....	4
2. Introduction	5
3. Background	5
3.1 Existing Solutions.....	5
3.1.1 Facebook.....	5
3.1.2 Instagram	6
3.1.3 Twitter.....	6
3.1.4 WhatsApp.....	6
3.2 Further Research.....	7
3.2.1 The Problem.....	7
3.2.2 The Solution	7
3.3 User Personas	8
4. Aims.....	9
5. Requirements	9
6. Design	10
6.1 Lo-fi Designs	10
6.2 Hi-fi Designs	10
6.3 Database Design.....	10
7. Development Plan	11
8. Language and Architecture.....	12
9. Implementation.....	13
9.1 Feed Screen.....	13
9.2 Trending Screen.....	13
9.3 Following System (Public Profile).....	13
9.4 Camera Screen.....	13
9.5 Events page.....	13
9.6 Private Profile	13
10. Approach	14
10.1 Database Revision	14
10.2 Experimental work	14
11. Testing	16
11.1 Usability Test - User Evaluations	16
Results:.....	16
Results:.....	17

11.2 Comparison Testing.....	17
Results:.....	17
11.3 Backend Testing - Performance measures.....	18
Results:.....	18
12. Limitations.....	19
12.1 Database Design.....	19
12.2 Application Design:.....	20
13. Conclusion.....	22
13.1 Future Work.....	22
14. Acknowledgments	22
15. Appendix.....	23
16. References.....	26

1. Abstract

Meeting and connecting with new people at university can seem a daunting task to many students for various reasons thus restricting them from fully experiencing all that university has to offer. The aim of the project is to create a safe space where students at the university can build a network of like-minded individuals and stay connected to information about what is going on in and around the campus and city. We achieve this through StudentPeak, a social networking mobile application running on both android and IOS developed with JavaScript and used firebase for the database and authentication. We believe StudentPeak can facilitate the formation of meaningful connections between students, by allowing students to find each other, keep in contact through messages, find events around the city, etc.

Through the report, we discuss the tools and factors that aided our product creation including the research undertaken and background for the idea, the development cycle undertaken to implement it, problems encountered along the process, testing carried out for the product and our analysis of the test results.

2. Introduction

With the age of social media rising now more than ever, it is being used as a prominent channel to communicate, connect and interact. It has become an integral part of our everyday lives, removing the struggles of communicating with each other. The rise of social media has transformed social interactions and the way in which people now meet each other.

Our objective is to create a social media app solely for university students that contains features that allows them to communicate, connect and interact. The features we will build aim to help new and current university students to connect with other students through their interests, explore social events at the university and discover societies and communities. These features have been decided through the results of our research - the insight in our research magnified into the struggles of university students, allowing us to develop features that will solve their problems.

The report outlines the development of StudentPeak, highlighting the methods we used to develop our features to build the app and successfully solve the problems of university students. We break down our process by firstly looking into existing solutions, the problems and solutions and user personas. We break down the tasks that are needed to complete the app, and go through our development process to implement the features to complete the app. We discuss our learnings with new software, the approaches we take and the problems we faced throughout the development stage. We also highlight our rigorous testing process to ensure the app is functioning correctly, ensuring that changes to the code have been made if tests have failed.

3. Background

We carried out background research to look into existing solutions, to look into what is currently available in the market and how StudentPeak solves a current problem.

3.1 Existing Solutions

3.1.1 Facebook

StudentPeak is a social media app, therefore our research was focused on existing social media apps appealing to students. When researching social media apps, we found that students had to use popular social media apps like Facebook as ways to connect with each other virtually.

Past studies have found how social media plays a huge role in allowing students to connect with one another, with "over 95% of British undergraduate students are regularly using social

networking sites” [1]. We have decided to take a look at Facebook, with the University of Kent’s Facebook having over 170,000 followers.

Whilst Facebook is a popular social media site for students to connect with each other, its sole purpose is not for university students, which StudentPeak is designed to be.

3.1.2 Instagram

Instagram is another popular social networking site popular amongst Gen Zs (who make up a large part of current university students), with Business Insider revealing that at least 62% of Gen Zs check Instagram daily [2]. Known as one of the most popular photo sharing apps, Instagram allows students who sign up to connect with their peers and share snapshots of their university life. Again, whilst Instagram is a popular app, it does not offer the niche requirements for its users to be students, meaning StudentPeak offers a unique proposition.

3.1.3 Twitter

Twitter is a social media app also used by Gen Z, with studies finding that almost 50% of tweets made between May 2020 - May 2021 were made by users aged 16-24 [3]. Twitter offers a platform for users to see and interact with trending content and share opinions with other users. A popular feature known as ‘hashtags’ originated from Twitter, a way to group popular content together to be interacted with by its users. StudentPeak will also aim to categorise popular content in the app. This popular content can be interacted with by students, who can keep up to date with what’s trending amongst their peers.

3.1.4 WhatsApp

WhatsApp is a social networking app used for direct messaging and group chats amongst users. With WhatsApp having over 2 billion active monthly users [4], it is a popular channel for digital communication. The app is used by university students, who often use the group chat function to interact with each other. Accommodation, course/module and society group chats are created by students as a channel to have the latest news on things happening. Whilst WhatsApp requires users to have a valid phone number to create a Whatsapp account, StudentPeak aims for university students to just have a valid university email (accessible to every student) to sign up for an account. This allows any university student to stay connected with their peers by using the app.

After researching existing social media apps on the market, we conducted further research to understand the experiences of new university students, in order for us to tackle the problem.

3.2 Further Research

3.2.1 The Problem

Upon arriving at university, students may find it difficult to interact and connect with their peers, making it tough to find friends. We conducted a questionnaire to ask current university students to ask about their experiences upon arriving at university as new students and to get an insight into their current social networking experiences at university. Our studies illustrate the problems current university students are facing, with *Figure 1 (See Appendix)* showing that over **30%** of our participants did not find it easy connecting with our students when arriving at university, and **50%** finding it relatively easy.

When asked if they had used any social media to meet other students, over **80%** (*Figure 2 in Appendix*) of our participants said that they had, indicating the importance of social media for university students.

We also asked if there was a need for a social media app solely for university students when other social media platforms like Instagram and Facebook exist. With **85%** of our participants saying yes, we asked them why. Responses to this question can be seen in *Figure 3*.

These results signify the need for StudentPeak, it will solve the problem many university students face, a problem that is not currently being solved by other social media platforms.

Additionally, to ensure we solve the problems that university students face when coming to university, we asked about the current features our participants use when using social media (*Figure 4*). With 'messaging' and 'sharing content' being in every response, we knew these were features we had to develop when creating StudentPeak.

See 'StudentPeak Market Research Report' file in corpus for detailed report.

3.2.2 The Solution

StudentPeak allows students to overcome the initial fear of being unable to find friends, providing them with a platform to gauge with students in the same year as them, studying their courses, living in their accommodation or even having the same interests. Social media plays a great role in connecting students with those around them, with studies suggesting that 95% of British undergraduate students use social media regularly [1].

The platform is primarily focused on connecting students in the University of Kent through its features, with the ability of finding societies to join and events to attend that are specific to students attending university in Canterbury.

3.3 User Personas

To aid our research, we decided to create user personas to provide insight into real behaviours and pain points of real users. By taking a look into the results from our questionnaire, we were able to create user personas from the struggles real university students face (*Figure 5*). We were able to look into the behaviours, goals, pains and needs of university students to create viable user personas. The aim is for StudentPeak to resolve the pains for users like our user personas.

4. Aims

The primary aim for our project is to help create a space for new students coming into university for the first time. We all experienced first-hand how hard it was to connect with people during the first few weeks of university, and our plan is to bridge that gap for the next set of students.

This drives us. We plan to make an application that helps connect students on a much deeper level, helping them to share experiences of their time in university and make new friends along the way. By having a single vision for the application, we can stream-line what features need to be included. We the developers, are also students, giving us an advantage as we look to create from that point of view. We want to make something unique as there isn't a social media application out there specifically for students and by creating this application, we would be a major player in the market.

Our report will show how we have gone about to achieve that, what we learned along the way and where we plan to go from this current stage of the project.

5. Requirements

Based on the aims behind the project we developed user stories to enable us to specify requirements to deliver a minimum viable product (MVP).

- StudentPeak shall allow user registration and profile creation
- StudentPeak shall allow user to register only with a Kent email
- StudentPeak shall allow user to view and edit their profile
- StudentPeak shall allow user to search for other users on the application
- StudentPeak shall allow user to follow other users
- StudentPeak shall allow user upload videos to their profile
- StudentPeak shall allow user to upload pictures to their profile
- StudentPeak shall allow user to view a feed of networks uploaded media
- StudentPeak shall allow user to search for events in their selected geographical location
- StudentPeak shall allow user to send chat messages to other users
- StudentPeak shall allow user to view other users' photos and videos
- StudentPeak shall allow user to view photos and videos of other users in their network

6. Design

As we worked alongside Group 1 to develop StudentPeak, both groups used the same user interface to develop the look of the app. The app's design had a minimalistic approach, allowing anyone to use the interface with ease and create a simple yet positive user experience.

6.1 Lo-fi Designs

Lo-fi designs were created in Adobe XD, which included basic designs of the features we were going to develop. This included designs for the user registration/login screen, the feed screen and the profile screen. Adobe XD allowed the lo-fi designs to include basic interaction with the use of basic buttons, which allowed navigation through each page to demonstrate the flow of the app. This was however still very basic and a rough outline of the app initially.

6.2 Hi-fi Designs

The Hi-fi designs were also created in Adobe XD. These designs were developed on top of the Lo-fi designs, and included more detailed designs of the user interface. It also included a higher number of interactions using buttons, allowing almost every element of the screen to be touchable and easier to navigate than the Lo-fi designs. Furthermore, the hi-fi designs illustrated how our requirements/UVP would be met e.g. The explore page designs showed how students could interact with other students they hadn't met before.

6.3 Database Design

We used Firebase to build the database and handle the backend data. Firebase makes use of collections (or tables) and documents which have fields. The fields in these documents contained relevant information about each user e.g. username, email, their degrees, year group etc. It also contained information about posts e.g. when the post was created, the type of post it was (picture or video), the like count, comment count etc. To illustrate how the database works and flows, we used Firebase's structure. By doing this it allows us to know how the data can be managed when we create/insert into the collections within our code. *See the Database Design file in the corpus to see in depth details of the database design.*

7. Development Plan

Agile methodology is a process in which a project is broken into different stages, and as the project develops over each stage, there are constant enhancements happening after conversations with team members and project leaders. The team decided to move forward using Kanban project management from Agile methodology. In line with Kanban, the team used the visualisation tool Trello to create a task board. The next step was to create a feature list document with more details about each task where one or more team members get assigned a task.

The feature list was created and hosted on Google Drive allowing all members to have access and view this in the form of a ticket. By doing this we were able to track progress. The feature list allowed us to view the status of a task, meaning other members could view the progress of other co-team members. A shared folder on Google Drive was created to store the necessary documents created over course of the project including the submitted corpus. We found its feature of easy sharing and access to access files from anywhere/ any device advantageous to our project.

As suggested in the Kanban Methodology, the team created a meeting log to record meetings, with each member filling a section on what they had just finished completing, what they are currently working on and if there were problems/other factors affecting the completion of that task.

Examples of the above can be found in our corpus folder.

7.1 Git / GitHub

The preferred method for monitoring the development cycle was GitHub.

Using Gits version control system, project members tracked as the code changes, which became extremely useful to the team as there were instances in which the code needed to be rolled back to a specific point before any changes.

Git has a hosting repository service called GitHub, which serves as our central repository, while also using the GitHub branch feature. We had the master branch where all completed code was stored. Each team member also had their own individual branch to work on. This allowed for collaboration across the project, as everyone can work independently, reducing the risk of human error.

Using merge, the project leaders could merge from master to individual branches after the latest code was tested and accepted to be fully working. It allowed team members to view the latest changes made to be code base as the project lead could push new iterations of the application across all branches.

8. Language and Architecture

The application was to be a multi-platform, using JavaScript (see next section) with its accompanying web technology Node.js. The framework of choice is React.js and the database to store images and user data is Firebase.

This section covers how Student-Peak was built and some of the influences behind some decisions taken.

8.1 API

8.1.1 What

Student-Peak uses React Native, a popular framework that boasts an extensive catalogue of libraries. Built using JavaScript, it follows the methodology of object-oriented programming; however, it does not force the user to use OOP techniques. The components created get reused across the application for different purposes. There are various examples of this across the application, e.g profile icon, profile search.

It was used to create the user interface of the application, allowing users to interact with different components within the application. For example, logging into your account or following another account.

8.1.2 How

Using the React view and state change capabilities, we can store information about different users and manipulate the state of the data as it changes every-time the current view is reloaded.

We used this with Node.js for back-end development to query Firebase. Firebase is a real-time database used to store and sync data between devices in real-time. It offers a REST API allowing programmers to query specific data from the database via the use of HTTP requests, allowing flexibility in our application to create functions that read, write & update from the database.

Following the specification of the Hi-Fi design, each screen created is a stateless component that calls a function allowing the data received to be re-rendered into the component view.

8.1.3 Why

React has a large community. This can be useful when we encounter a problem beyond our scope. It also uses JavaScript XML or JSX that allows you to declare HTML syntax directly in JavaScript code. This was useful, as the team members had previously used JavaScript/HTML in the first 2 years of computer science.

9. Implementation

9.1 Feed Screen

The feed serves as the homepage of the application. Once logged in, it serves as a page to interact with users they follow, by viewing what's recent and interacting by leaving a like or a comment. To view content, you need to be following at least 1 one user. At the top of the screen, the user can choose to view only pictures or videos.

9.2 Trending Screen

This screen allows users to view trending topics and posts and interact by liking or commenting. It also serves as a page that allows you to search for a specific user or a topic and view the results returned.

9.3 Following System (Public Profile)

Integrated within the search component on the trending screen, this allows a user to search for another user by querying the firebase database. If they exist, the screen displays their public profile, providing an option to follow said user. This was further integrated into the feed screens components as when you click on the profile image/username, it calls the search function returning the profile of the user.

9.4 Camera Screen

A component allowing the user to share their media. Using the device's camera or selecting media from their device. They pick a media type (photo/video), advance to the save screen to add a caption and upload the post. This is uploaded to the database under information related to the userID and new associated data such as comments and likes is saved in the postData collection, this information. This was initially created by our team but was then passed off to G1 to add additional features.

9.5 Events page

This screen allows the user to view current events, presented with options to choose their preferred event type, after this the user can pick their current location and submit this, the submit button queries the Google Event Results API and returned results are displayed in a new page providing graphical information about events.

9.6 Private Profile

This page displays the profile information of the currently logged in user, it shows information such as the number of people currently following the user and an option to log out of the application. There was a collaboration with G1 to work on some aspects of this screen.

10. Approach

10.1 Database Revision

As the project moved through various iterations, we learnt more about what can be done with the React and the Firebase API by experimenting with different ideas and changed the content of the database to meet our current needs.

An example of this was the associated data for a user's post, which had to be changed a few times to meet the users' requirements. At the start of the project, it only contained information about its storage location in the database and what type of media the post is.

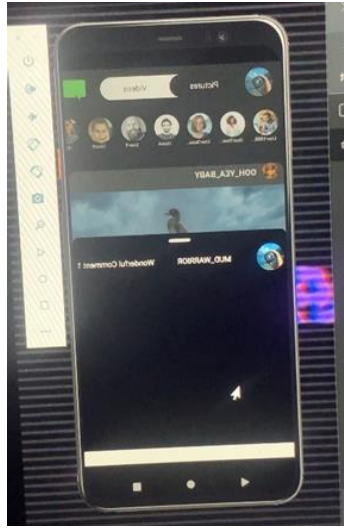
As we moved forward with the task, it was apparent that users would like to view more information about posts being displayed. Firebase does not use pre-defined schemas, instead, it uses key-value or document stores. A new collection housed the associated data created and connected using the post as a key to store its values.

There were other scenarios in which we had to change our approach. Sometimes what we created on the database had to reflect in the code when querying and vice versa.

A Database diagram has been included on the corpus titled Database Design.pdf.

10.2 Experimental work

There was very limited experimentation, as was following the specifications of the HI-FI design. The comment section was not included in the design requirements, giving the team free rein on how it gets displayed. We experimented with different ways to show the comments on a post, explored different display modals before deciding on having the comments on a new page when the user clicks on the icon. See below for comment experiment.



11. Testing

We carried out a number of tests to analyse StudentPeaks performance and to gain feedback on the app. Another aim for testing was to ensure our requirements and specifications that we set at the beginning of the project were met.

Our testing included usability testing to gain user evaluations, backend testing to evaluate the app's performance, and comparison testing to compare our app with other products that have the same purpose as ours.

11.1 Usability Test - User Evaluations

To gain appropriate feedback on the application, we carried out usability testing. We used our phones to run the app and would give this to the students. Once they had a go at using a particular screen, they would answer the questions based on that feature in the questionnaire.

We presented the app and the first set of questions to a random group of students. These first set of questions were primarily based on the app's user friendliness, asking the user testing the application if the features we developed were easy to navigate without us guiding them. The questions also focused on the app's user interface, asking the user how easy or difficult they found using the app based on its interface. We also asked user's to provide their thoughts about the design and layout of the app. This was to gain insight into the user experience, and to ensure our requirements are met on the user interface.

Results:

Based on the results of the usability testing, we found that users testing the app found it easy to navigate and understand. See *Figure 6* for responses to the question **“What are your thoughts on the design and layout?”**

With **100%** of the users answering 'Yes' to **“Is the system easy to use?”**, we see that the user interface and experience resonates positively with our prospective users. Whilst we had a mostly positive response to the application, further improvements can still be made to the application from the responses we got (*Figure 7*)

The second set of usability questions were asked to test the quality and usability of the app, and to provide further insight into the views of who would be using the app - university students. Again, we asked fellow students to use the app and provide us with feedback based on their experience using the application's features. We used the same method of running the app on one of our phones and presented it to a random group of students, who would firstly use/test a screen and then answer questions based on that screen.

The questions in this test primarily focused on the likes and dislikes on the features we built and asking the users to provide feedback on improvements for these features.

Results:

From the responses we gathered, we found most to be positive. When asked if users could use the features on the Events, Explore and Feed screens, **100%** of users answered that they could.

These responses demonstrate our requirements being fulfilled, with users being able to interact with new students through the explore screen, find events to go to using the events screen and stay connected with their friends through the feed screen.

We were also provided with useful feedback on how to improve the app (*See Figure 8*).

These responses help give us an insight into work that can be carried out in future work and improvements that could be implemented if we had more time.

The full test report can be found in the 'Usability Testing' file in the corpus.

11.2 Comparison Testing

Furthermore, we performed a comparison test to compare our application to similar applications on the market to highlight the unique features that distinguish our product from others. This test was carried out in two parts. Firstly, we tested our application's functionality against known standards of successful social networking applications to ensure that our application meets the market requirements. We then perform some specific feature tests to assess and identify our application's unique value.

Results:

The results of the comparison testing revealed that certain features we implemented met the standard requirements of successful social media apps currently on the market. Moreover, we could also conclude StudentPeak offers unique value that other apps on the market do not currently offer e.g., the events feature provides students with events happening in their university city/area whereas Facebook provides a search bar for people to search for events they already know about.

Where users have to search for specific trending topics within other competitor applications, StudentPeak collates a list of posts with the most interactions within a certain time period and displays this to the user in the explore screen under a 'trending posts' label.

A feature that we acknowledge can be improved from the results of the test is displaying the search results of tagged posts in a more orderly form.

The full test report can be found in the 'Comparison Testing' file in the corpus.

11.3 Backend Testing - Performance measures

To test the backend of the app, we carried out backend testing. This included using the app's GUI to test the features that were using the database to ensure the database was being updated, deleted from, read from or created (e.g. new records were being created and inserted into tables). As the database was being hosted on Firebase, this is where we would inspect the data. We carried out various tests - e.g. when a post is created by a user, the post's table would be updated. Carrying out backend testing also gave us insight into the performance of the app and how the backend handles the data. *The full test report can be found in the 'Backend Testing' file in the corpus.*

Results:

When testing the database, we found that most features functioned appropriately with the database. If a test for a feature failed the first time, we would go back to that feature to ensure it passed the next time we tested it. However, there were certain features that we tested that failed the first time but did not end up implementing or continued to fail using the GUI. An example of this was deleting comments, a feature that was able to work when using the database (we were able to delete comments using Firebase) but not the app's GUI - one of the limitations of the app.

Performance wise, we found that the more data that was stored in the database, the slower the app became to load data. This was another limitation that we found as we continued to test the app with increasing data, with a solution that could be found in future work.

12. Limitations

12.1 Database Design

One of the limitations this project has is to do with the database design and how data is structured within our projects database being firebase.

For example, the users collection which is made from the following path format; “/users / \$uid”.

The user’s database is a collection which is populated with documents which are named after the userID correlating to the user the document is about. This document is then populated with fields regarding information about the user.

```
{
  "users": {
    "userID"...": {
      "email" : "...",
      "password" : "...",
      "name" : "...",
      "surname" : "...",
      "dob" : "...",
      "gender" : "...",
      "anonymoususername" : "...",
      "username" : "...",
      "course" : "...",
      "yearofstudy" : "...",
      "stage" : "...",
      "bio" : "...",
      "topics" : "...",
      "friends" : "...",
      "photos" : "...",
      "messages" : "...",
      "accomodation" : "...",
      "stayeround" : "...",
      "nationality" : "...",
      "placeofstudy" : "...",
      "followers" : '0',
      "following" : '0',
      "videos" : "...",
      "likes" : "...",
      "societies" : "...",
      "profileimage" : "...",
    }
  }
}
```

As seen above in the table called users which stores all the information regarding a user’s profile such their: email, username, nationality...

In the scenario of querying data from firebase, a developer cannot just fetch specific fields from a document when needed. However, the whole document must be fetched which returns a document with all its fields.

The problem this poses is that this database table although is intended to encapsulate and provide all of the data regarding a user in one place - this database collection is majority of the time being constantly used by the feed screens to fetch only specific fields being a; user's profile image and username from a userID (2/25 fields provided) to display information about the user who made a specific post when rendering the feed screen.

In terms of efficiency the feed screen is requesting a lot of unnecessary data per user on the feed screen to acquire the 2 fields of data it needs regarding a user who made a specific post from their userID.

In my opinion the user's database structure should be split into multiple different databases which provides precise information required for different processes because there isn't a single process in the application that requires all these fields of data to be returned altogether.

For example, the user's database could be split into different database collections which follow the same structure as the original users database but, contains different fields in their documents. The database collections as such would be:

1. User Personal Information – which stores documents containing the users: dob, email, nationality, surname and any other personal information.
2. Profile info –which stores documents containing the users: username and profile image which is required by the feed screen.
3.

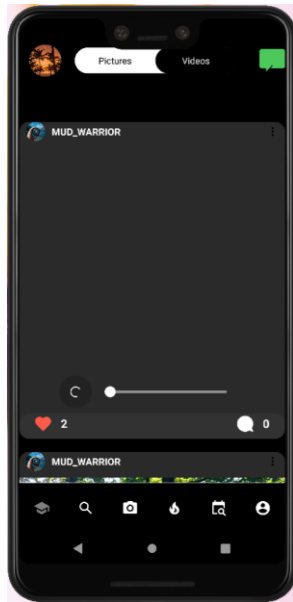
Additionally, besides efficiency a slight security flaw is that a user's email shouldn't be returned in the backend of the application to another user who's feed rendering the feed screens as this information could be extracted by hackers.

12.2 Application Design:

Another limitation of the application is the screen design in regard to the feed screen.

The feed screen is a screen which has nested inside of itself 2 screens being the picture, video feed screens which are then controlled by a switch selector.

The switch selector on the "Feed Screen" controls whether the "Picture Feed Screen" or the "Video Feed Screen" is visible. Once the switch selector is set on the "Pictures" option the "Picture Feed Screen" is then made visible on the "Feed Screen" and vice versa.



One problem that arises having 2 these screens being nested onto a single screen controlled by a switch selector is that when the state of the switch selector is changed, the requested screen is then re-rendered with the data it was originally passed through its flatlist which means the changes such as liking a post are reverted back.

To combat this, we designed the likes and comments object to request new data every time this object was being re-rendered so, that this object is up to date with the current likes/comments count. This isn't efficient as each time the screen the feed screen switches between picture to videos each time the screens are re-rendered each post has to request from the DB the likes status, count and comment count for that post.

13. Conclusion

We set out to create an application that will aid networking among students at the university and our final product achieves a great number of our core requirements. StudentPeak is capable of improving students' university experience by allowing them to connect with other students through posting comments, sharing media through content upload, searching for and following other users and finding events to attend within the city. Catering to a niche audience (university students), we believe StudentPeak will have a place in the market due to some of the unique features e.g., informing students of local events, etc, which distinguish it from similar applications in the industry.

Upon reflection we have learned about what developing a fully functional application entails and take a different approach to certain areas of development were we to re-do this project. Taking the time constraint to complete the project into consideration alongside other commitments to our degree, we would have fewer features that require a steep learning curve so as to enable us to deliver a fully functional result that can be improved on. Overall, we are happy with the results of our project as it signifies a year of learning, failures and turning failures into lessons to achieve a goal.

13.1 Future Work

We acknowledge that enhancements could be made to our application and plan to further feature implementation to StudentPeak after the deadline to increase the application's viability.

An initial feature discussed among the team at early stages of the project is a student bubble, which would use a matching algorithm to connect users based on certain chosen criteria such as their course, interests, etc. Unfortunately this feature could not be implemented due to the time constraint on the project. Introducing a student bubble would be beneficial to the users as they would have a variety of options to decide how they want to build their network and who to add to it.

Student Peak is currently a cross platform application that runs on both IOS and Android, however, creating a web application version would increase its reach and accessibility to the target audience.

14. Acknowledgments

We want to say a massive thank you to our project supervisor Kemi Adeyoke. Thank you for pushing us and guiding us throughout the project, your advice each week helped us immensely. We would also like to thank all participants who took part in our surveys and questionnaires.

15. Appendix

Figure 1, Results from questionnaire

How easy did you find it connecting to people in your first year?
31 responses

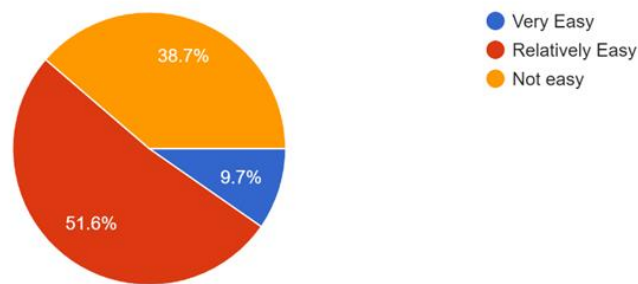


Figure 2, results from questionnaire

Did you use any social media apps to meet other university students?
31 responses

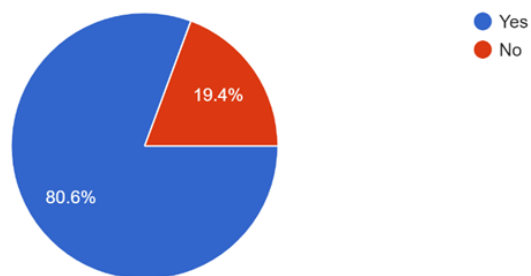


Figure 3, results from questionnaire

As much as facebook and instagram are fun social media platforms, it is hard to connect with a specific group of people e.g someone doing your course or living in the same student accommodation without having to know specifically where to look and know somebody that knows about it first. By having an app like this it would ensure you find your people much quicker and and it wont be as lonely!

So you can know that everyone's who's on the app has the same intention of meeting others as you which you would be guessing if you approach someone on other social platforms

It would be nice to connect with other students that are not only on my course

To target a niche (uni students) and it would be easy if it's solely for uni students usage

Figure 4, results from questionnaire

What features of any social media platform do you use when connecting with peers? (e.g. messaging, sharing content etc)

31 responses

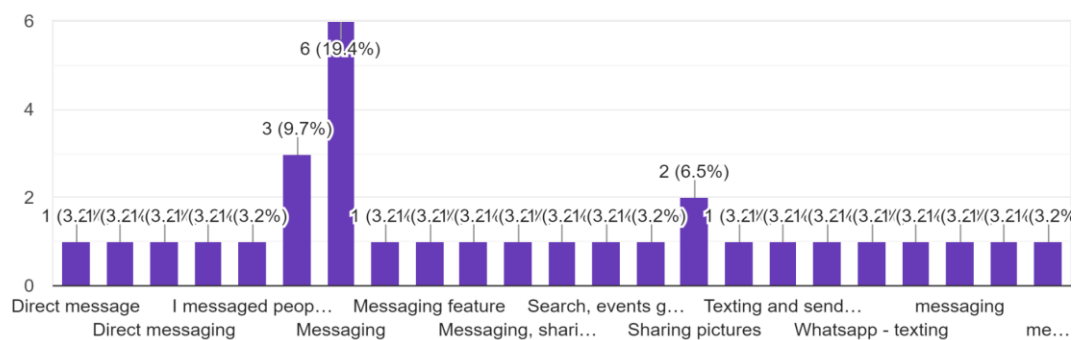


Figure 5, User Personas



Luke Smith

Luke is a 19-year-old first year student who studies Economics. He is interested in music and football.

Behaviours

- Spends time with friends
- Plays music with his band
- Plays in football club

Goals

- Wants to make friends on his course
- To join an Economic based society

Needs

- To find people in his accommodation to go out with
- To find other students who play music

Pains

- New to living alone and being independent
- Adjusting to not being around his friends from home
- Unable to find the right friends that will have the same interests as him



Isabella Montez

Isabella is a 23-year-old postgraduate student who studies Digital Marketing. She is interested in art and travelling.

Behaviours

- Solo travelling
- Going to art classes during the weekend
- Interning at a marketing agency

Needs

- Finding postgraduate students that live in her accommodation
- Find an art society she can join

Goals

- To make friends who enjoy travelling
- To connect with other students who are Also interning whilst studying

Pains

- Adjusting back to university life after graduating 2 years prior
- Finding events that are targeted towards postgraduate students

Figure 6, Usability Testing responses

What are your thoughts on the design and layout?

7 responses

It was clear

Nice and simple design, easy to get used to.

Good and understandable

it's very simple and easy to understand. It gives me what I need and provides me with enough substance to engage with the app.

Easy to understand and navigate

Layot is clean and clear, makes you want to use the app. Maybe a bit of colour would add.

Figure 7, Usability Testing response

Things are a bit disproportionate, and scaled weirdly. A lot of information on the page. Other than that it's good.

Figure 8, Usability Testing response

Feed Screen: What can we improve about this screen?

6 responses

Maybe include when the photo was posted

To be able to save pictures

See who liked your pictures

Make it more clear when one person's post finishes and another posts ends.

the speed

User experience

16. References

1. Madge, C., J. Meek, J. Wellens, and T. Hooley. 2009. "Facebook, Social Integration and Informal Learning at University: It Is More for Socialising and Talking to Friends About Work than for Actually Doing Work."
2. Dennis Green. 2019. "The most popular social media platforms". [Accessed April 2022.] <https://www.businessinsider.com/gen-z-loves-snapchat-instagram-and-youtube-social-media-2019-6?r=US&IR=T>
3. <https://www.socialmediatoday.com/news/twitter-shares-new-insights-into-gen-z-usage-and-engagement/608193/> [Accessed April 2022]
4. <https://www.keystoneacademic.com/news/whats-up-with-whatsapp-how-the-app-is-changing-campus-dynamics> [Accessed April 2022]