

PERSONALISED BUZAPP IOS

CO600 PROJECT
TECHNICAL REPORT

M.JEAN-PIERRE (MJ314) AND SAHIB JABBAL(ssj9)
UNIVERSITY OF KENT

Abstract

BuzApp is a bus application that is intended for University of Kent community to use, as they travel around campus and the surrounding areas. Prior to this project, BuzApp offered core features such as when the next bus was due to arrive, but this project builds on this and adds personalisation features.

1. Introduction

BuzApp is a transport application for the iOS platform that was earlier created by the previous year with the core features for the University of Kent community allowing the community to travel around Canterbury. Our objective was to successfully personalise the application which could allow the community to use the application that is personalised for their needs. This is to allow the community to have a more engaging experience with the application while they are on the move.

BuzApp kept the simple interface that was created by our predecessors as it allows for swift transitioning between various features of the application. This application allows the user to set their frequent used bus stops to favourites, which can be retrieved within a short period of time. Similarly, the recents feature stores all the bus stops that have been explored by the user. Journey Planner

is another feature of BuzApp that will allow the user to provide to and from locations and provides them with the steps they will need to take on the journey to ensure they get to their destination most efficiently. BuzApp also provides a calendar feature which integrates the iOS calendar into our application and can calculate the journey you need to take to reach the location of the event.

This report outlines all the work that we undertook throughout this project, starting from the planning and research stage to the end product where we have an application with a multitude of new features. In the initial stages of our project, we first looked at what the previous group had done and provided a critique for this, before researching into the Transport API, personalisation and what similar solutions are doing. After carrying out this research we planned how we wanted to structure the project, where we took an agile approach and started to outline the features that we wanted to include. Developing was the next stage and we split the work into iterations where we aimed to deliver features in each of these stages. Once these features had been implemented, we carried out testing to ensure they worked as planned.

2. Background

Prior to starting the project, some background reading and research was required for us to carry out to allow us to have better understanding of how best to personalise the BuzApp application. This involved understanding how the previous group implemented each feature, the tool they used to gain the bus information and how personalisation is currently being carried out and how this can be adapted to our project.

2.1 BuzApp iOS Critique

To begin the project, we carried out some research into the BuzApp application that was previously created and wrote a critique about it (App Store, 2017). This was a very important task for us as we would be building on the project that was already created so was very important to not only understand what the previous group had done but also how we could improve upon it. This allowed us to have an initial think about some of the features that we can look to implement and have a look at the framework for how this could be technically possible.

2.2 Transport API research

The critique allowed us to see that the main way data is collected through the app which is via the Transport API. The prior group

performed REST GET web service calls to access the data that was needed. This was important for us to understand as it was the foundation of the application that they created and how this data is retrieved and placed in a presentable form for the user. This also allowed us to look further at the Transport API for the additional features that they support and if it would be possible to use this information for personalised features (Transportapi.com, 2017).

2.3 iOS Development

To develop an application on iOS, swift programming language is used on the XCode platform (Developer.apple.com, 2017). Swift is currently on version 4 and uses a storyboard design as the core element of creating the user interfaces of the application (Developer.apple.com, 2017). This is the first time either of us had developed in swift, hence this was a steep learning curve.

2.4 Research into Personalisation

The main goal of this project is to add personalisation features to the BuzApp application to be utilised by our target market which is University of Kent community. Therefore, we conducted our research into personalisation to find the true meaning of the features that we are planning to implement. Additionally, we also looked at examples of how other travel

applications are being personalised and even looked beyond our market to how the best-personalised apps on the market are personalising the user's experience. Some examples of applications that we looked into for personalisation was TripIt, Waze and CityMapper to list a few (App Store, 2017). The outcome derived from the research we conducted was that there are two types of personalisation which were content and experience. Based on these definitions we knew to focus our features around these types of personalisation which were to change the application to meet the needs of the user or make it look the same for each user and have the content in the application change per the user.

2.5 Research into Similar Solutions

BuzApp application is in the travel category thus, we conducted research into other applications in this area as they would be our competitors. We targeted bus apps that are widely used in Kent and even some that are used in London to get the similarities and differences for applications based on the area that they are used in.

2.5.1 Stagecoach

Stagecoach are our key competitors as they provide the main service in Canterbury and other areas of Kent. Through carrying out our research we found that the main features of their application were nearby

stops, journey planner and adding a home and work location. This was very interesting as they have encapsulated some personalised features such as journey planner and storing locations for each individual user (App Store, 2017). We took some of the features in consideration for BuzApp that we want to implement for this project.

2.5.2 Bus London

The Bus London app had similar features to Stagecoach such as journey planner and the ability to store favourite stops. The favourites feature is very interesting as it allows the user to personally store the stops that they use regularly for easy retrieval (App Store, 2017). After researching both the apps we had a good idea of the type of features that we wanted to implement to not only match our competitors but potentially overtake them with the content that we will provide.

3. Requirements

The primary requirement for this project was to personalise the BuzApp application for the University of Kent community. We aimed to satisfy the needs of the community by adding features that would be beneficial to the users with a simple user interface so that it's easy to navigate amongst the

features, these are explained in our user stories.

3.1 User Stories

During the preparation of the project, we created a list of user stories that were used to help us in the implementation of the personalised application with the use cases.

- As a student at the University of Kent, I want to be able to add my frequently visited stops to my favourites for easier access
- As a student at the University of Kent, I want to be able to view the recent stops that I have searched for, to gain quick retrieval of the bus stop.
- As a student at the University of Kent, I want to be able to add my start location and end destination so that a calculation can be made on the steps needed for me to reach my destination.
- As a University of Kent student, I want to be able to get the events from the calendar for the day, to allow me to plan my journey with the journey planner.

3.2 Non-functional requirements

As well as the functional requirements of the application, we designed non-functional requirements that the application must have, some of which are listed below.

Efficiency

It is vital for the application to be efficient for the user, therefore the app must be developed in order for the application to run as efficient as possible to allow all the key data to be displayed to the user.

Response Time

The application is built on the foundation of data about buses and bus stops, therefore it is important that the information provided to the user is fast and allows them to be up to date with the buses.

Correctness

The application needs to provide the correct data to the user. This is because our application is predicated on real-time information that could impact a user's journey. Therefore, we need to make sure that the content they are receiving is accurate and up to date.

Usability

The application needs to incorporate the simplistic interface that was incorporated by the previous group. This will allow the user to seamlessly navigate through the application without getting confused.

4. Development Iterations

4.1 Iteration 1

The focus of this iteration was to be able to have a great understanding of the programming language (Swift) that we

would be using to implement the personalised features in the next iteration onwards. While understanding Swift our aim was to rectify the transport API calls from the previous year's project and use our app id and key that is provided by the Transport API.

4.1.1 Features

API Retrieving Information

The aim of this iteration was to rectify the transport API calls as the data required wasn't available to us hence we were required to register to the transport API as developers in which an app id and app key is generated for us to use in the JSON calls and use our Swift understanding from research to solve the transport API data.

4.1.2 Challenges

Our main challenge during this iteration was understanding the features of the API and how the API is used to provide the data and being able to understand the structure of the code that was written by the previous year's students as we were building on the application.

4.1.3 Results

The first iteration gave us the background understanding of Swift and confidence in our ability to develop new features to the existing application. In addition to gaining

more knowledge, we were able to fix the transport API calls to get the desired data.

4.2 Iteration 2

The focus for our second iteration was to be able to add new personalised features to the application from our research. We decided that we are going to implement the favourites and recents feature.

4.2.1 Features

Favourites

The main feature that we focused on in this iteration was the favourites feature. This would allow the user to add their frequently used bus stops to their own personal list for them to be able to retrieve this information quickly. This would be done by clicking on the bus stop and allowing the user to personally add to their list. We also allowed the user to remove the stop from their favourites. This gave us our first hands-on approach to developing in Swift and allowed us to gain a lot of knowledge.

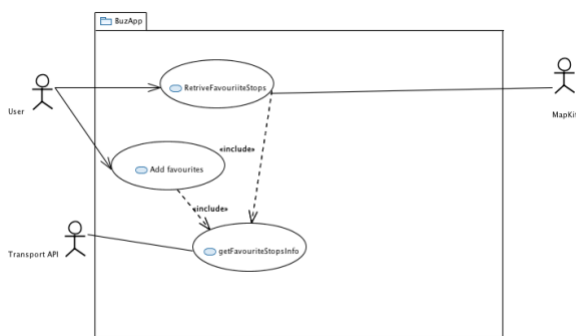
Recents

Our next feature in this iteration was to implement a recents feature. This feature is to populate a table of bus stops for the user based on the recent stops the user has clicked on. The user would be able to find their recent stops and be able to view the recent stop from any location as they might want to use the bus stop at a later time

which may not be their nearest bus stop at the time. The user is also able to remove the recent stops from the list for stops they do not need.

4.2.2 Software Design

Prior to implementing the features that we agreed on, we created use case diagrams. This was used to give us an understanding of the relation between the user and the application, which helped form the code that was used to implement these features.



In addition to the use case diagrams, we also created a class diagram for this iteration. The class diagram helped us develop the functionalities of the features as it guided us through the iteration implementation process.

4.2.3 Storyboard Design

To design the new features, we needed to modify the current storyboards in XCode as it would allow the users to interact with the new features. We added new screens by placing buttons, table views, labels and

setting up the connections between the screens. The storyboard feature was an essential tool for us to develop the new design of the application efficiently.

4.2.4 Challenges

The main challenge that we faced in this iteration was populating an array with information from another class. As we had to take the information about the stop and send this to a favourites class, we had difficulties but managed to overcome this. This was done by creating a global array in the destination class, which then allowed us to pass the data from one class to another, populating the table view with this information.

Our next challenge that we faced was displaying the information about the bus stop once the stop was clicked from the favourites tab. As we were sharing a controller that was used for the nearest feature, there were some issues with this allowing us to display the correct data. We managed to overcome this challenge by sending the values needed to the bus stop view controller. We also created a separated segue between the two classes which fixed this problem.

Our next challenge during this iteration was pre-loading the favourite and recent lists of

added favourite and recent bus stops when the application is not being used and is opened again. We overcame this challenge by understanding how user defaults are set and used.

4.2.5 Results

This was a difficult iteration as we had a few problems and was our first real experience having to implement something in Swift. However, we were able to implement the favourites and recents feature and the features worked as we intended. We found that once we had implemented the favourites feature, recents was quite easy to implement as it built on the same principals.

4.2.6 Testing

After completing the iteration, we created some tests to be carried out. The following tests were carried out;

Unit Tests

These were carried out to test the core functionalities of the application ensuring the desired data was retrieved. This test was carried out to test that the application is able to get the favourite stop. This was done by passing a preset value to the Transport API and making sure this works as intended. A similar test was carried out to get the direction of the bus, setting a preset value and making a call to the API. These exact

tests were then changed and tailored to the recent view controller for getting the stops and direction. All the unit tests passed as expected.

User Interface (UI) Tests

UI tests were carried out to test the interface of the application ensuring the application is working smoothly. This was carried out for favourites by selecting a stop, adding it to the favourites and then ensuring that the list was populated with this correct value. Once this had been completed, we deleted that stop from the favourite and checked that this was represented on the list. We also carried out UI tests for recents which similarly checked that the recents stop was populated with the correct value once a user had clicked on a stop. All of the UI tests for this iteration passed as expected.

User Acceptance Tests

The last form of testing that we carried out in this implementation was the User Acceptance Tests. We split these into three rounds throughout the implementation to make sure that the application was working as intended in its environment where it will be used. The testing involved not only testing the new features that we implemented, but also the features that were carried out by the previous students. We did have a failure for favourites in the second round of testing. However, we were

able to fix this by the third round and all features passed.

4.3 Iteration 3

In this iteration our aim was to implement additional features individually giving us more responsibility as we had some experience working with Swift. Having gained some experience using Swift we were confident that we could implement the features individually, hence we distributed the features between ourselves.

Like our previous iteration, our purpose for this iteration was focused on adding more personalised features to the application.

4.3.1 Features

Journey Planner

One of the features that we were looking to implement in this iteration was the Journey Planner. This allows the user to implement the start and end location and gives them the steps they need to take to reach the destination and how long the overall journey and each step will take. This would be linked to Apple Maps which would take the locations given and display the route in a map view.

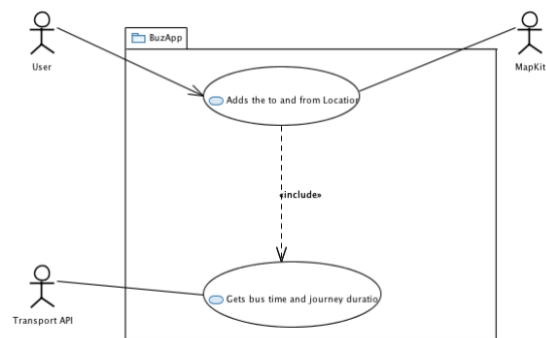
Calendar

The calendar feature is integrated with the calendar on the user's device. It takes the events that they have for the day and allows

them to plan their journey to get to this event on time from their current location, integrating with the Journey Planner feature.

4.3.2 Software Design

Similarly, to iteration 2, prior to implementing the features, we created use case diagrams to plan how the software will



be designed and how all the different parts interact with each other.

We also created a class diagram for this iteration to guide us through the implementation process for the iteration

4.3.3 Storyboard Design

To design the storyboard, we had to first look at the use case and how we would want the user to interact with the system. As we were trying to keep the UI consistent, we added the bare minimum elements to make sure the user can still fully utilise the feature. Based on this search bars for the user to enter data and table views were used to present the data that the user needs.

4.3.4 Challenges

These were two big features of our application and there was a multitude of challenges that we faced throughout the iteration. The Journey Planner had issues that came with gathering the data from the API. The data provided by the API was in JSON and JSON parsing is required to be able to retrieve the data in Swift. Another challenge that the Journey Planner encountered was that there were issues due to the data being in nested arrays via the API. Additionally, once the JSON data had been parsed, it was difficult retrieving this data and presenting it on a table view.

The calendar feature also had some challenges throughout the implementation. One of the challenges faced was getting the events from the calendar for all of the available calendars on the phone as the user may have different sources of calendars on the device such as the subscribed calendar from the University. There was also a challenge of getting the accurate location of the University of Kent lecture and seminar rooms. As the API only deals with main locations around the country it wouldn't recognise the location based on just the seminar or lecture feature. Therefore, an alternative solution was required to be implemented to solve this problem.

4.3.5 Results

As a result of this iteration, the Journey Planner and Calendar features have been implemented. The user is able to manually search two destinations and be given a calculation of the time and steps needed to take to reach their destination. The user is also able to integrate their calendar with the application to pick an event that they are planning to go to and get a calculation on how to reach this destination on time.

4.3.6 Testing

Unit Tests

These just like in iteration 2 were carried out to test the core features of the application, getting the desired data. This test was carried out to test that the application is able to get the location that the user has inputted. This works by sending a pre-set location to the Transport API and making sure that this returns a result. Additionally, to this, a test was carried out with two pre-set locations and a call made to Apple maps was made to ensure that this worked as expected. All unit tests for this iteration passed as expected.

User Interface (UI) Tests

UI tests were carried out to test the interface of the application ensuring the application is working smoothly. This was carried out for the calendar feature by testing that the event location retrieved is sent to the

Journey Planner. All of the UI tests for this iteration passed as expected.

User Acceptance Tests

The last form of testing that we carried out in this implementation was the User Acceptance Tests. This, just like iteration two was split into three rounds. In the first two rounds, we decided not to test the previous functionality as the work we are doing doesn't interfere with these features. Therefore, in the first two rounds, it was strictly for the Calendar and Journey Planner implementation. There was only one problem with the Journey Planner in round two as it pertained to calculating the journey with the from and to locations. In round three everything for this iteration had passed and we also tested all of the other features in round three just to double check that our implementation didn't affect another part of the application.

5. Agile Development

We decided to develop this application using the Agile Software Development method as while developing mobile applications, it is important to break it down into sections to allow for a simpler visualisation of the goals that we have. We used Agile in this project by breaking work down into iterations. Therefore, at the start of an iteration, we had the main goals that

we were focused on. This worked well as we produced work in each iteration and allowed us to reflect at the end of each iteration to see what we could improve on in the next one. Also, it allowed us to adapt and change between iterations if this was necessary.

6. Quality Assurance

We conducted Quality Control on all our documents and source code that we completed. For every document we completed we ensured that the document was reviewed and completed the Quality Assurance Form. Each iteration of our code was evaluated to make sure there was consistency throughout the code and easy to understand.

6.1 User Acceptance Tests

Along with the unit and UI tests performed, we also conducted a user acceptance tests in which we performed user acceptance tests for each iteration of the application. This allowed us to get a view of how a user would use the application for the first time without a user manual.

7.Evaluation of Project

7.1 Goals

The main goal of this project was to add personalisation features for the BuzApp

application for use for the University of Kent community and surrounding area.

From the start of the project, we always had this goal in mind, which allowed us to tailor the work we carried out towards this. This is demonstrated throughout the research that we carried out at the beginning of the project, and the user stories that we created and adjusted throughout the iterations. Based on this I feel like we carried out our main goal of the project. We were able to implement four different features, tailored to the person and they all work as intended.

7.2 External Evaluation

To justify that we had met the goals of the project, we carried out some external evaluation. This was done by giving our finished application to a number of people from the University of Kent community and gaining their thoughts from this. We gained all positive feedback from the members of the community asked, with a common theme being the ease of use of the application and how it was able to store the personal user information that they had added to the application based on their activity. We were feeling really positive about this as it justified that we had reached the main goal that we had set out at the initial stage of the project.

8. Conclusions

8.1 Project Overview

This project was put forward for us to create a personalised bus app for the University of Kent community and surrounding area. It built on the BuzApp application that had previously been created, adding more personal features that would appeal more to our target audience.

8.2 Final Product

As a result of this project, we have added personalisation features to the existing BuzApp application. The features include favourites, which allows the user to select a bus stop that they commonly use and add it to a list for convenience. Similarly, we have added a recents feature, which based on the user's activity, will store the bus stops that they have recently clicked on, which once again allows for quick retrieval. The calendar is another feature that we implemented, which takes the events for the day from the user's iOS calendar and passes this to the final feature, which is the Journey Planner. This feature allows the user to search for their from and to locations and then calculate the journey. This will then open the Apple Map with the locations selected and default to the transport section for the user to immediately start their journey.

8.3 Future Improvements

8.3.1 University Calendar Integration

One improvement for BuzApp in the future would be for the application to link directly to the University of Kent calendar. As this would be more commonly used by the community it may be more convenient for them and could still be linked to the Journey Planner feature.

8.3.2 Journey Planner

An improvement to the journey planner is to be able to add a favourite home or work address that the user can be able to plan a journey using the addresses.

8.3.3 Push Notifications

Another feature that we had implemented in the next iteration was could be push notifications. This would be a feature that sends a notification to the user about when the next bus is arriving to the stop that they are currently at. This would allow for the users experience to be more personal and present the information in a more convenient manner.

8.3.4 Inbox feature

This feature would allow the users to view any messages regarding their planned journey. For example, if a bus is late, a message would be sent to the user. This would be integrated with the push notifications feature

9. Acknowledgements

We would like to thank Rogerio De Lemos for supervising our project, giving us advice and making sure that we are put in the best position to do well. We would also like to thank Apple for not only giving us the platform to develop but also through their maps integration, helped with the features we managed to implement. Lastly, we would like to thank the Transport API for being the main source of data for our application and making sure that all the information we receive is up to date and accurate.

10. Bibliography

- App Store. (2018). *Waze Navigation & Live Traffic on the App Store*. [online] Available at: <https://itunes.apple.com/us/app/waze-navigation-live-traffic/id323229106?mt=8> [Accessed 9 March. 2018].
- App Store. (2018). *Bus Times London on the App Store*. [online] Available at: <https://itunes.apple.com/gb/app/bus-times-london/id480643687?mt=8> [Accessed 10 March. 2018].
- App Store. (2018). *BuzApp on the App Store*. [online] Available at: <https://itunes.apple.com/gb/app/buz>

app/id1206897388?mt=8

[Accessed 8 March. 2018].

- App Store. (2018). *Citymapper Transit Navigation on the App Store*. [online] Available at: <https://itunes.apple.com/gb/app/citymapper-transit-navigation/id469463298?mt=8> [Accessed 9 March. 2018].
- App Store. (2018). *Stagecoach Bus on the App Store*. [online] Available at: <https://itunes.apple.com/gb/app/stagecoach-bus/id1151110474?mt=8> [Accessed 10 March. 2018].
- Developer.apple.com. (2018). *Swift - Apple Developer*. [online] Available at: <https://developer.apple.com/swift/> [Accessed 8 March. 2018].
- Developer.apple.com. (2018). *Xcode - Apple Developer*. [online] Available at: <https://developer.apple.com/xcode> [Accessed 8 March. 2018].

gecoach-bus/id1151110474?mt=8

[Accessed 10 March. 2018].

- App Store. (2018). *TripIt: Travel Organizer on the App Store*. [online] Available at: <https://itunes.apple.com/gb/app/trip-it-travel-organizer/id311035142?mt=8> [Accessed 9 March. 2018].
- Developer.apple.com. (2018). *Swift - Apple Developer*. [online] Available at: <https://developer.apple.com/swift/> [Accessed 8 March. 2018].
- Transportapi.com. (2018). *Transport API – the digital platform for transport*. [online] Available at: <https://www.transportapi.com> [Accessed 8 March. 2018].

10. Appendix

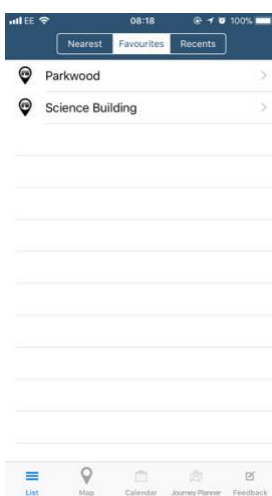


Figure 1-Favourites

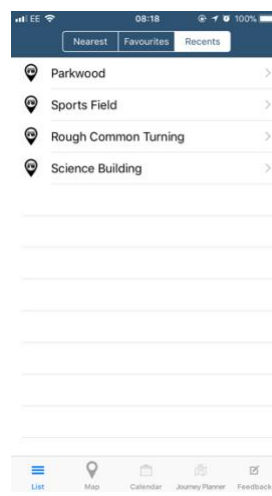


Figure 2- Recents

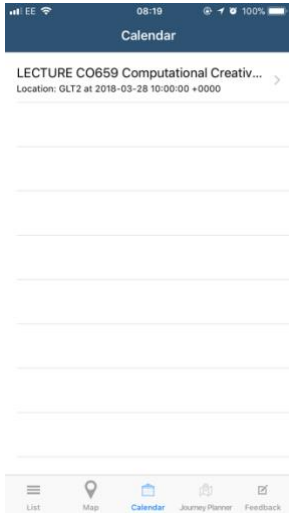


Figure 3-Calendar

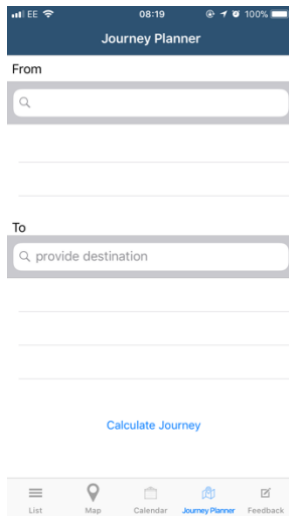


Figure 4- Journey Planner