**TRAVEL PALS**

33439825, Jessica Suszek , [jsusz001@gold.ac.uk](mailto:jsusz001@gold.ac.uk)

33402940, Pedro Branco, [pbran010@gold.ac.uk](mailto:pbran010@gold.ac.uk)

33439088, Tanmim Hanifa, [thani001@gold.ac.uk](mailto:thani001@gold.ac.uk)

33433114, Mohamed Bhailat, [mbhai002@gold.ac.uk](mailto:mbhai002@gold.ac.uk)

Goldsmiths University of London

Smartphone application that uses social media to encourage people to travel together.

**Table of Contents**

[Introduction 3](#_Toc473106138)

[Development Post-Mortem 3](#_Toc473106139)

[Formative Evaluation 3](#_Toc473106140)

[Design and Implementation 3](#_Toc473106141)

[Quality Assurance 3](#_Toc473106142)

[Summative Evaluation 3](#_Toc473106143)

[Bibliography 3](#_Toc473106144)

[Appendices 3](#_Toc473106145)

# Introduction

Travel Pals is an Android smart phone application that uses social media to allow and help people communicate and meet with others sharing a common travelling plans or destinations. This report will show how this was developed, tested and evaluated. We plan to achieve is ideally a fully finished product with some thorough testing as well as meeting all our set requirements discussed in the proposal of this project. In order to determine the success we will have to evaluate the feedback and usability by users, however everything depends of how we will distribute the team work within the work force as well as how steady our progress in the implementation of the app.

# Development Post-Mortem

We have started the term with a plan that we have decided on last term (discussed in the proposal- page 20 see the gantt chart). The job roles would have been split as such, we would have two teams; team one, responsible for the implementation and team two responsible for the report. Our strongest programmers were supposed to be Arman, Mohammed and Tamin , however due to Arman’s unexpected leave , we have to restructure our team. Arman was supposed to be the main programmer since he had previous experience creating phone application which would have been very helpful since the rest of us are all novice programmers. Unfortunately, due to his departure we had a major setback starting.

We have naturally set of a waterfall development methodology, trying to following the grant chart seen in the proposal. We tried to achieve each functionality one by one which meant that we all must sit and wait until one step is completed. Due to the time restriction on this project, we had to revalued this approach since a lot of time was being wasted. In order to manage the project development more effectively and efficiently, we have set on an agile development methodology. Since we are all novice developers, we had to first focus on producing a simple working prototype. This also gave us an ability to change functionalities as we were developing them each sprint.

The tasks to be done were managed by having a visual product backlog and a sprint backlogs for each sprint. The project backlog was a prioritised list of all the tasks to be done during the project (different functionality). The sprint backlogs were like the product backlog except they were prioritised lists of tasks to be done within each individual sprint with each task having a group member of assigned to them that would be responsible for performing that task.

This figure displays the product backlog of each sprint that we have taken and what we have achieved.

The project was divided into a set of 5 sprints. Each sprint would be conducted over the course of 1 and a half weeks, totalling to 7 and half weeks. Since we had a delay starting our sprints were shorter in order to meet the deadline. At the beginning of each sprint a meeting would be held to conduct the sprint planning. During the meeting the group would decide the two important roles for the sprint, the tasks from the product backlog to go into the next sprint backlog, who should be responsible for which tasks, and to review progress since the beginning of the last sprint. we would have two meetings to discuss the progress of each sprint and any problems.

We have had a little technical hiccup, since one of our programmer Mohammed was struggling to fix a bug in the “View plans” page of the application, since it displayed all the users with all their plans rather than only displaying the plans for that user logged in. This has delayed us a bit since other functions such as the search depended on the plans and used the same programming logic. However, we have managed to overcome this by doing some extensive research as well contacting a university collogue who had programming experience in Studio Android.

Another issue we have faced as a team, near the end of the term, we all have been faced with a lot of pressure from other courses therefore the meeting attendance has been effected, and that caused miscommunication and lack of quality in the presentation.

Overall, the project was successful since we have overcome any obstacles that came our way however, there is room for improvement. Next time, when approaching such project it’s very important to have a very good strategy and follow up with it as often as you can in order to keep up with the progress and acknowledge and solve any problems that may arise.

# Formative Evaluation

We have attempted to do testing to an extend for every functionality every sprint that we had, to help up identify any user-based problems, of the application and so we can add it in our backlog too. This was done by both verification and validation techniques of every functionality.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test functionality** | **Testing** | **n of users** | **Test cases** | **Outcome** |
| Registration form | Verification and validation of the data fields that the users enter information in. | 5 | * Enter special characters * Enter duplicate data * Leave all field empty | * Duplicate registration * Special characters entered * T&C missing * 18+ tick box validation missing * lack of error messages |
| Login page | Verification that the user exists | 5 | * Leave field empty * Enter incorrect password * Is the password visible when entered | * incorrect password (error message) * forget password option * lack or error messages |
| Menu | User friendly look | 5 | * each option takes to the correct page * matching icon for each page | All users said that it’s easy to use and don’t need explanation of how to use it |
| View Plans | Show all information, and display clearly | 10 | * see all plans * delete plan * edit plan | Users understood what they got back, however, they didn’t know how to delete a plan or edit a plan |
| Add Plan | Validation and verification that fields entered are displayed then in View Plans | 10 | * enter special character * add empty fields | Users found that anything such as location can be added to the in the text fields which is not a location. A suggestion would be to have a drop-down option |
| View/Edit Profile | Edit information about the user | 5 | * edit into an empty field * add special characters | * lack of error messages |
| Simple Search (by destination only) | Display users by their destination plans | 10 | * enter not a location | Users would like a more visual view to see the users description more clearly. Too much text on one page |
| Advance search |  | 10 |  |  |
| Messaging | Can send messages |  |  |  |

In the table above shows the static testing that has been done on each functionality, on how many numbers of end- users and the outcomes we received. We have done this by conducting a quick observation of how random users around the campus used the app and gave up verbal feedback.

This type of testing was the easiest, in the time frame as well as users were more willing to conduct this rather than spending several minutes on questionnaires etc.

Also, when developing each functionality, we tried our best to bring our personas alive and did some functional; and non-functional l testing in accordance, before giving it to our test group. This was done in order to avoid any small easily fixed bugs and issues such as; password has to be hashed, or the view plans page displayed all plans from all users, such technical bugs in the code was identified early.

Another aspect that we had to look at is the database, as we tested each functionality shown in the table above, we have to check the database whether everything is changed accordingly as we expected to.

# Design and Implementation

Give an overview of your final design and implementation, including any changes from your initial ideas. Provide justifications for any significant decisions you made.

Given the wide scope of the our user requirements summarise in the proposal last term, we have discovered a lot of elements to our application that we did not even take into consideration when prototyping at first.

**Android app**

This was a very simple requirement to meet since we have been head set to create an android application using Android studio, however we did consider translating our idea into a web application instead since we had been faced with technical difficulties that we thought we might not managed taken in consideration the time restrictions.

**Messaging functionality**

**Search/match functionality**

We have managed to create a simple search function, by location only . Meaning that the user only enters the destination and other user’s plans show up, the result in a view list. Our visual aim when displaying this result was to have each plan combined with some user profile information in a full page view , and then the end user can swipe to see the next result. The difficulty in that was , we used a JSON library which produced a result in a certain format which was hard to change when received from the PHP file to Android Studio. Another issue with this functionality is that, who comes first and why? How do we base if which result is the best result to display first, in order to avoid this dilemma we have implemented a list view result set, however realistically , it would have been nice to have an algorithm that would get the results list searched by destination and then sort them out perhaps on different factors such as profile pictures (studies show that users are more inclined to speak to other users if they have a real profile picture), description , a comparison between each of the users plans looking specifically at the dates and only display the users at the top with a close match to the user’s plan. These little things, would make sure that the end user gets the best possible result first time round when looking for a companion , therefore proving the aim of the application right and hopefully come back to use it again and again .

We had also attempted to implement a advanced search which would conduct a more specific search, not only on the user plans but also on the user profile details such as gender and age range, in our proposal our researched has shown that our target audience preferred to have such a functionality since they can be more exact on their own preferences when travelling, for example a young female would not enough travelling with an older man , since she may not feel entirely safe as well as their trip might focus on a different aspects such as partying rather than visiting museums. In order to achieve this , we have created a more detailed search looking specifically at age range, gender, destination , date of travel

. This makes the search more exact therefore users might not receive as many results. However an ideal advanced search would preferably have a set of default values therefore, if a user misses to enter or select a specific field, they would still be able to search and receive results.

**Swipe like motion for a match (like tinder)**

Due to implication we had developing the search and the plans functionality , it had delayed our project therefore we did not managed to have a go implementing a swipe like function to more from result to result after conducting a search , instead we have settled for a list view , which is still a simple to see and users can compare visually each result.

**Sequential navigation**

We have settled on a simple menu page which has 4 navigation icons; it’s sequential. After creating each page that was needing for the application we have to create button and upload an icon, as well as attached a link to a page that it’s going to be connected a redirected to. There is not drop down menu, since we decided that on a small screen it looks very crowded, therefore we have created a menu button which takes you to the menu page wherever page you are at.

**Add trips functionality**

When implementing this function we have realised we need two pages , one to display all the plans created from our user and one for the user to actually add their plan. In order to do this , when the user clicks on the plan button it gets directed to the a page where it shows all the currently existing plans by the user in a list view , where the user pereferably is able to delete and edit each plan.

There is a small button which directs the user to another page where it displays a form to add all the fields to create a plan/trip. Once the user “adds” the plan , it data is set to the database in via an PHP file which transforms the data is a certain format and then enters the new record accordingly to the SQL INSERT statement.

The problem that we have encountered was trying to received the data from the database. Till then we have been using a JSON library which has helped us format the data when inserting , however to retried the data by the user ID that has been logged was a bit harder than anticipated. We have attempted to do the same as before with the insertion of data , but just simply changing the SQL statement to SELECT , however this showed the results only in the PHP file , we know this by using the Postman extension which was very helpful indentifying the problem . So we were stuck at Android studio since we have to make the ID a global but private variable which can be accessed in this view plans page. Etc …

**Profile settings**

**Login screen**

Regarding our ethical audit in the proposal , we have mentioned that TravelPals will have an option to login with facebook, however with further discussion this term with our target audience and peers we have discovered that information from the facebook account that would have to be retreieved when logining in via facebook , would majority be redundant. For example, TravelPals, only takes in one profile picture , therefore is no need for access to all the pictures that the users in mentioned in , location , and all the extra private details are not necessary therefore users are better off registering with just their email , name , age and password fields . If we have added a facebook functionality , it would mean that as owner of the application we would be responsible and liable for all this personal redundant to us information about the user which can be misplaced and misused as well, therefore to avoid we have decided to hold back on this option. In addition it is mentioned in the Data Protection Act, see the UK government website , that we can only hold “adequate, relevant and not excessive” meaning that data that we do not need , we cannot get a hold of or keep it.

**Blue colour theme**

We have tried to keep our design as similar as possible to our final design last term, also this made the design much easier when implementing since we knew what was wanted exactly and where. The blue theme was definitely achieved.

# Quality Assurance

Provide details of your approach to QA and any testing carried out, including results. How well does your final system conform to your initial requirements? Justify any changes in requirements.

**Quality of Design**

In term one , we have produced very detailed prototype of our application, which we tried to match when implementing this application. Few things had to be changed due to implications mentioned through the report and the time restriction. However the overall design and look of the application does match over prototype requirements showed in the proposal.

**Quality of Conformance**

As per our requirements stated in the proposal, we have managed to meet majority however, due to the time restrictions and several implications we have failed to meet the following:

* messaging functionality between users
* login with facebook
* swipe like motion when looking through users

We have also found that there has been some ambiguity and lack of precise requirements, therefore the implementation of serval functionality were not to everyone visions since the requirements did not clearly underline these specifications. Nevertheless, we have found space for improvement when conducing user testing which had ensure a certain level of quality within our software. The feedback received was crucial since it was the main source that we could evaluate the progression and the quality of our application, it has also underlined bugs that we could fix before our final release.

Also, due to our lack of experience when conducting a project of this magnitude and the absence of knowledge prior starting the implementation phase about Android Studio and the compatibility between these different platforms: Android Studio, SQL, PHP and other libraries. Has caused a problem when integrating all functionalities together and the quality of the code when it will come to continuance of the project.

**Software testing**

We have used the debugging functionality in Android Studio as well as Postman extension in order to identify any issues with our code. Android Studio debugger showed any complication when complying the code and Postman extension showed if the SQL queries where correcting fetched via PHP and how they were formatted for Android Studio to retrieve and then display.

**Testing Objectives**

We have attempted to test our software by running through “Scenario Tests”. See the Formative section of this report for more details. Each scenario had to be run through the delegated functionality to verify whether all features are working as expected, this includes error detections and messaging, the navigation linkage, correct results displaying etc.

**Verification and Validation**

As mentioned above we have conducted thorough testing through running our application through different scenario tests as well as some user testing. Also we have worked collaboratively in this project using GITLAB which allowed all team members to commit their work and others to view and identify and verify any confusions as well as check if it was in conformity with the requirements set.

**Testing Methodology**

Non function

# Summative Evaluation

Describe the methods, results and conclusions of the evaluation of your final software.

# The overall deliverable of this project was sufficient to the expectation we have set ourselves. This means that we are happy with our progress, however there is a lot of room for improvements. The application is fully functioning with the expected functions, however more advanced functionality is missing such as the swipe like motion when displaying all the results when conducting a search, or the forget password verification system that would be meet security requirements.

Even though we have received positive feedback from our peers, we have also received further suggestions of how we can develop this simple prototype further. Such as a good messaging functionality and as well as a block function of any abusive users that mistreated the code of conduct, uploading pictures of what user would like to see in the my plans functionality to make it more visual, as well as displaying notification of the phone if someone messaged you or when you have been matched. In order to make the matching more credible for users it would be ideal for the users to be matched from both sides or perhaps when a user is matched a request would be send to the matcher. The advanced search could be more efficient and useful if it had default values set in order to conduct etiquette searches. Another improvement could be to implement a trip tracker, so other optional details could be added if the user would like to add the flight details or exact location of the area of where they will be staying, also this idea can be expand to connectivity with companies such as Trip Advisor to improve and encourage users to leave feedback on their “past” trips, therefore we could be generating profit afterwards that way as well as advertising any top travel deals. Another tip to keep the database more up to date , would be to write an algorithm that would compare the current date and any trips to see if they have already went by, which means the trip has been expired and would be more temporarily to past trip which then can be disposed are a given time. This would make the database more efficient and integrational, as well as reduce expenses on storage.

There are plenty of ideas for improvements and expansion on this application, however on the time restriction given and given our experience we have tried our best to produce a final application to the best of our capabilities. Also, given the implication we overcame during this process we have managed to create a good base of an application that has a lot of potential.

# Marks distribution

We wish for the marks to be shared equally between all group members.

# Bibliography

<https://www.gov.uk/data-protection/the-data-protection-act>

# Appendices

TESTING

**Documentation file has to be added**