**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.

|  |  |
| --- | --- |
| **Requirements** | **Justification** |
| Android app | Refer to research in questionnaire 1 |
| Messaging functionality | Refer to research in questionnaire 2, users enjoyed communicating with each other |
| Search/match functionality | In order to find someone, you can talk to and plan your trip, we need to find the best available match |
| Swipe like motion for a match (like tinder) | Easy functionality as well as highly enjoyed by Tinder users |
| Sequential navigation | Makes the application more enjoyable by having an order navigation paths |
| Add trips functionality – user can add any upcoming trips | Keeps track of any travelling plans the users in wanting to making, and makes it easier for the database to find matches |
| Profile settings – adjustments and ability to add more information e.g. add photograph | Editing profile is crucial since the user might want to update some personal information about themselves |
| Login screen with possible registration or login with an existing account from Facebook | Refer to research in questionnaire 2, users prefer to sign up via facebook, since saved time and restricted user input |
| Blue colour theme | Refer to research in questionnaire 2, users related travelling with a blue theme |

# 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.

# Summative Evaluation

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

# Marks distribution

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

# Bibliography

# Appendices

TESTING

Description and motivation of development methodology

• Evidence of planning and regular reviews

• Systematic testing regime including design of appropriate test cases

• Justification of testing methods

• User evaluation involving representative stakeholders

• Insightful discussion of results

• Evaluation of own work in relation to original proposal and plan

• Reflection on what has been learned about coordination, negotiation and communication within a group.

• Outcomes should be evaluated with reference to efficiency, correctness and suitability to users’ needs.

• Ethical audit: you should explain how you have handled any issues of privacy, data protection, intellectual property…

**Documentation file has to be added**