

Android App for Local Taxi Bookings

| | |
|----------------------|-------------------------|
| Report Name | Project Outline |
| Author (User Id) | Rhys Evans (rhe24) |
| Supervisor (User Id) | Chris Loftus (cwl) |
| Module | CS39440 |
| Degree Scheme | G400 (Computer Science) |
| Date | February 7, 2019 |
| Revision | 1.1 |
| Status | Release |

1 Project description

The Android App for Local Taxi Bookings project will primarily involve creating a mobile app that is intended to serve as an intermediary between local taxi companies and their customers. It will likely support three types of users: Company Admin, Driver and Customer. All users will have to register an account, the app will have access management to ensure only desired features are available to different classes of users.

Customers will be able to make bookings through the app and provide information such as location, number of passengers, time and additional notes (i.e instructions for entry to tricky pickup locations). The customer can then either specify a desired taxi company or select 'no preference'. A push notification will be sent to the customer whenever the status of their booking changes, they will also be able to check this within the app whilst their booking is active. Customers will be able to cancel their booking at any time and view all their previous bookings.

Company Admins will have access to several management tools/settings. These management tools will allow Company admins to set the 'available hours' of their taxi service, add or remove drivers and block or unblock customers. Company admins will also be able to select how bookings are delegated and managed by the app. For example, if they select automatic delegation, the app will delegate bookings to drivers using an algorithm similar to line hunting in telephony. However, if they select manual delegation all bookings will come through to the company admin, who will then manually delegate bookings to drivers.

Drivers will 'belong' to a given taxi company that is managed by a company admin. They will be able to mark themselves as available/unavailable within the app; being 'unavailable' will exclude them from automatic delegation. If the driver's company has automatic delegation selected, they will be able to Accept or Decline bookings as they come through. Drivers will also be able to mark the status of their current booking, which will be sent to the customer. Depending on company preference, the live GPS location of the driver could also be broadcast to the user whilst the booking is active. The app will have a 'driving mode' option available to drivers when enabled it will present a more simplistic and easy to use UI.

The main goal of the app is to streamline the process of making taxi bookings, both for customers and for taxi companies. A history of previous bookings can be made available to each customer and taxi company. Such a history could be extremely helpful for cases of lost property, criminal damage, etc. Taxi companies would also be able to identify and blacklist problematic customers. By viewing the live status of their booking, customers can also make sure they are present for their booking at the correct time, hopefully resulting in fewer taxis waiting around.

2 Proposed tasks

In order to complete this project, the following tasks will need to be completed:

- **Document the chosen methodology.** Extreme Programming [1,2] has already been selected as the chosen methodology for this project. However, decisions still need to be made regarding the practices of Extreme Programming [1,2] that are best suited for this project. These decisions will need to be documented for reference throughout the project.
- **Identify key functionality and create user stories.** This work will involve identifying what the key, high-level requirements of the project are and writing them down in the form of user stories. Each story should have some sort of acceptance test outlined for it.
- **Look into build processes and deployment platforms.** This task will involve researching continuous integration and deployment solutions such as Jenkins [3], Travis-CI [4] and Team City [5]. Options for hosting the Application's backend server will also need to be considered.

- **Research the technologies and languages required for the Application's backend.** This work will involve researching options for data persistence [6], and various different server-side technologies, such as Node.js [7], Express.js [8] and Ruby on Rails [9], and perhaps more.
- **Setting up a Git Repository and local build environment.** This task will involve creating a Git repository and setting up a local development environment that is compatible with the aforementioned development tools.
- **Development and Integration of the Backend Server.** Information security [10] will need to be constantly considered in order to develop a robust, reliable and GDPR [11] compliant application.
- **Development of the Android App.** This aspect of development will focus on the Android App itself. The look and feel, identity and branding of the app will need to be established. The user interface of the application will need to be designed and prototyped [12, 13]. The app will also need to interact with the backend. Automated tests will need to be created for each feature of the application.
- **Write the Final Report.** The final report will contain all documentation relevant to the built system.
- **Prepare for Demonstrations.** This task will involve making preparations for the project's demonstrations. These will involve ensuring the system to be demonstrated works as intended and prepare a verbal presentation to accompany the technical demonstration.

3 Project deliverables

The following deliverables can be expected:

- **A brief document outlining my chosen methodology and coding standards.** A brief document will be produced to describe and justify the choice of methodology. This will include a short breakdown of each practice that is part of the methodology and a section addressing the coding standards of the project.
- **Story Cards.** The story cards created as part of the chosen methodology will be documented, there will also be a brief description of the acceptance test for each story within the document.
- **An online blog documenting weekly progress.** This deliverable will be continuously updated, up until the week of submission. A plain HTML/CSS blog will be created and made available to the project's supervisor.
- **Final Completed System.** The final system will contain a fully functioning Android App with an aesthetically pleasing user interface and a fully functional backend. The app should meet all the criteria outlined in Section . 1. This deliverable will contain all source code and a planned demonstration of the system.
- **Test suite.** A suite of tests that will be submitted alongside the final system. They will cover all automated tests and perhaps some manual tests that are created throughout the app's development. Tools such as Espresso and JUnit will likely be used to create the majority of tests.
- **Final Report.** This document will contain the project report in its entirety, including associated appendices. In this document, all 3rd party libraries, frameworks, and tools used in development will be formally acknowledged.

Initial Annotated Bibliography

- [1] K. Auer and R. Miller, *Extreme Programming Applied: Playing to Win*. Boston;London: Addison-Wesley, 2002, 0201616408.

This book is part of the XP Series, it is described as a pragmatic guide to getting started with Extreme Programming. This book will provide guidance on how to apply the important principles and practices of XP in this Project.

- [2] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change*, 2nd ed. Boston: Addison-Wesley, 2005, 0321278658.

This book provides an explanation of the practices and principles of Extreme Programming and gives an insight into the philosophy and motivations behind its creation. The book is authored by one of the original creators of the Extreme Programming methodology.

- [3] Jenkins, "Jenkins," <https://jenkins.io/>, Feb. 2019, accessed February 2019.

Jenkins is an open source continuous integration and continuous delivery tool in the form of an easily configurable server.

- [4] Travis CI, "Travis ci," <https://travis-ci.org/>, accessed February 2019.

Travis CI is a continuous integration tool that can be synced with github projects to create a build flow.

- [5] JetBrains, "Teamcity," <https://www.jetbrains.com/teamcity/>, accessed February 2019.

Team City is another CI/CD tool for consideration, it is created by Jet Brains; the same company who develops IntelliJ Idea, the platform on which Android Studio is built.

- [6] MongoDB Inc, "mongodb," <https://www.mongodb.com/>, accessed February 2019.

MongoDB is a free and open source NoSQL database solution. It is document-oriented and uses JSON-like documents.

- [7] Various, "Node.js Foundation," <https://nodejs.org/en/>, Feb. 2019, accessed February 2019.

Node.js is an open source JavaScript runtime that can be used serverside. It will be considered as an option to build the REST API needed for the mobile app.

- [8] Node.js Foundation, "Express.js," <https://expressjs.com/>, Feb. 2019, accessed February 2019.

This is a web framework for Node.js, it will be considered as an option to build a REST API together with Node.js

- [9] D. Hansson, "Ruby on rails," <https://rubyonrails.org/>, Feb. 2019, accessed February 2019.

Ruby on Rails is a web application framework for server-side use, it is written in Ruby. It will be considered as an alternative to Node.js to create the REST API and backend services needed for the App.

- [10] K. Duuna, *Secure Your Node.js Web Application*. Dallas: The Pragmatic Bookshelf, 2016, 1680500856.

This book provides guides and best practices to improve the security of Node.js web applications. It provides tips for creating robust database calls, user authentication, and session management.

- [11] “Data protection act 2018,” <http://www.legislation.gov.uk/ukpga/2018/12/contents/enacted>, May 2018, accessed February 2019.

This act controls how personal information is used by organisations, businesses or the government. It is the UK’s implementation of the General Data Protection Regulation (GDPR)

- [12] Adobe, “Adobe xd cc,” <https://www.adobe.com/uk/products/xd.html>, accessed February 2019.

This is an user interface design and prototyping tool made by Adobe, there are material design expansions available. These can be used to quickly create extremely accurate Android Application prototypes.

- [13] Fluid UI, “Fluid ui,” <https://www.fluidui.com/>, Feb. 2019, accessed February 2019.

Fluid UI is a web-based user interface design tool. Similarly to Adobe XD, it too has material design UI packages.

- [14] Mills, C. et al, “Express web framework (node.js/javascript),” https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs, Dec. 2018, accessed February 2019.

This is a user-created guide to using Node.js and Express.js, it is a resource that would be very useful if Express.js/Node.js were used to build the REST API.

- [15] C. Sevilleja, “Express web framework (node.js/javascript),” <https://scotch.io/tutorials/build-a-restful-api-using-node-and-express-4>, Apr. 2014, accessed February 2019.

This is a tutorial for building REST APIs using Express.js, it provides an introduction to routing using Express.js 4 and the use of middleware.

- [16] Okta, “Okta authentication,” <https://developer.okta.com/>, Feb. 2019, accessed February 2019.

This is an authentication API that would be used to implement user authentication for the App. It is considered as an alternative to building user authentication from scratch in Express.js/Node.js (perhaps for prototyping)

- [17] Docker Inc, “Docker,” <https://www.docker.com/>, accessed February 2019.

Docker allows for containerization of software. Containers are isolated and can be used to bundle tools, libraries and configuration files for simple and easy deployment. Docker works very well with CI/CD tools.