Department of Computing
Goldsmiths, University of London

Augmented Reality Navigation System for Commercial Spaces

Deployment Plan

by

**Arif Kharoti, Nicholas Orford-Williams, Hardik Ramesh,
Gabriel Sampaio Da Silva Diogo, Hamza Sheikh, Jonathan Tang**

Software Projects – Group 14

Version 1.0
Spring 2019

1 Introduction

1.1 Purpose

The purpose of the deployment plan is to ensure that the system successfully reaches its users and new features to the system are delivered successfully. The aim of the deployment plan is to provide a detailed schedule of events, persons responsible, and dependencies required to integrate the new version of the app with the previous version. It should minimize the impact of the integration of the new system on the users and stakeholders.

1.2 Assumptions

The application would have a place for users to:

- Have a login activity, where the user can enter credentials to login, or continue as a guest.
- To login to the application, enter their location, and destination.
- Route calculation takes place (finding the shortest route to the destination).
- The user can retrieve their current location, and map.
- The user can take a quick snap of a exhibit and the application provides more information about it.

1.3 Dependencies

Dependencies that can hinder or slow the process of deployment are:

- Dependent on using an Android device
- Dependent on Google's ARCore Software Development Kit (SDK)
- Dependent on Google's Map services
- Unavoidable change of plans
  - Change of user requirements
  - Research fails
  - Implementation fails
- Time management

  - Group meeting
  - Supervisor meeting
  - Weekly delegated tasks
  - Milestones

1.4 Constraints

- Reliance with Google's map services (server can crash)
- Repository could be down
- Group member availability:
  - Not all members are mutually available
  - Conflicting time schedules
- Internet Connection/Wi-Fi issues
- Database error: Existing accounts may not be able to login if the DB server is down

2 Assessment of Deployment Readiness

1. Verify that the application does not have any broken links and that all content is accessible.
2. Verify that all dependent files have been uploaded to the relevant directories so that they can be accessed from other calls.
3. Supervisor approval

2.1 Product Content

Configuration would include the following:

- Accuracy and reliability of separation of programming plans by members.
- How easy to download all the documentation, report or code from gitlab.

2.2 Deviations and Waivers

Deviations from the original plan included:

- Implemented Route Calculations using Wi-Fi instead of Bluetooth
- Changed our 3D Model from a directional arrow to a navigational line
- Implemented outside commercial spaces instead of within a museum setting - to deliver applicable concept

3 Phase Rollout

Phase I

- Map showing user's current location
- Route calculations
- Superimposed 3D directional line
- Display navigation

Phase II

- Show nearest museums to the user's current location
- Camera recognition of exhibits
- Request and pull information about exhibit
- Display the information

Phase III

- Account database
- Registered users can store their visited museums
- User can rate and review the visited museums

4 Notification of Deployment

After the application is successfully released, a notification will be sent to stakeholders and clients. All iterations of the system will be detailed in the changelog.

4.1 Steps

1. Check all procedures and ensure everything is done.
2. Email client for meeting.
3. Present the client with the application.
4. Sign off development plan documents.
5. Email client with application information.
6. Release of project and approval of supervisor.

5 Deployment Systems

Continuous Integration and Continuous Delivery (CI/CD) will be used to deliver any changes to the system. Automated tests are written to for each new feature to ensure that less bugs are passed to the production stage and captured early by regression, reducing the risks at every release. Gitlab have built-in tools to support CI/CD, which provides a simplified setup and execution of software development using continuous methodology.

6 DevOps

Much like Agile development, adopting a DevOps culture allows for smoother processes within development.

- **Building the right product**: After code has been written, the team will be able to receive faster feedback as a result of live testing.
- **Improved productivity**: As delivery would be continuous, the developers and testers will be additionally efficient as testing environments are easier to set up (see A/B Testing below).
- **Reliable releases**: Smaller and more frequent releases would allow for less changes made to the code as well as the bugs.

6.1 A/B Testing

The main idea behind A/B Testing is that testing can be performed on variants based on live environments. Since development followed the agile methodology as well conforming to a test-driven development approach - testing was performed in conjunction with the coding. After new code has been written and when testing is required, the environment would already be set up for current and future tests.