Contents

1 Introduction

1.1 Purpose of this document

This document is designed to capture and discuss, the key processes and outcomes of the Doubtfire Helpdesk Ticketing System. This document will outline (1) how the system is designed to function, (2) how the system will effect stakeholders, as well as to assist the designers and developers of the system in the development of the project. It should be a reference of the development methodology chosen, which is outlined in the SDLC plan document.

Disclaimer: This document serves as a *guide* to the development process which will be conducted throughout this final year project. Granted the agile methodology adopted and user-centric design process, there may be significant alterations between the content outlined in this document and the final outcome itself. This is an understandable consideration considering that requirements will change as development continues, given the feedback provided from the client at the end of sprints. This has been understood by the client and product owner of the system.

1.2 Background

1.2.1 Swinburne University Programming Helpdesk

The Programming Helpdesk has been offering programming assistance to students in their first and second year of programming for many years. Over time, as the number of subjects supported by the helpdesk grows, the helpdesk has become busier, and as a result, it is very difficult for tutors working at the helpdesk to keep track of who they have seen, who still needs help and so on.

1.2.2 Doubtfire Learning Management System

Doubtfire is an open source learning management system currently in use across multiple subjects at Swinburne University of Technology and other universities in Australia. It's used by many staff and students on a daily basis. It provides students a simple and easy place to manage their unit, manage where and when they upload work, and is the place in which they receive feedback from their tutors for their submitted work.

1.3 Key Project Personnel

1.3.1 Client

The client for this project is Andrew Cain, as he oversees the running of the helpdesk and is the administrator of Doubtfire at Swinburne University. He is also a primary collaborator of Doubtfire.

1.3.2 Stakeholders

- The **Project Client**, Andrew Cain (`acain@swin.edu.au`)
- The **Project Supervisor**, Graham Farrell (`gfarrell@swin.edu.au`)
- **Teaching staff** and **students** in any unit that utilises Doubtfire as the primary learning management system used for marking and providing feedback to students.
- **Swinburne University of Technology ITS** which hosts Doubtfire and maintains server-side hardware.

1.3.3 Project Manager

Andrew Cain (`acain@swin.edu.au`) is the product owner and also a primary developer to the Doubtfire learning management system.

1.3.4 Project Members

Refer to the Group Contact Details[1] of this portfolio.

2 Terms of Reference

2.1 Goals

The Doubtfire Helpdesk Ticketing System has very clear goals which were arrived at through use of the system, and through interviews with the client. The system is intended to provide an simple, non-obtrusive way in which students can create a "ticket" when they are physically at the helpdesk, at this point tutor's should receive seamless notifications regarding their updated tickets, so that they know who and when they need to assist students who have open tickets.

[1]See `https://github.com/final-year-project/documentation/wiki/Group-Contact-Details`

The system needs to be integrated with Doubtfire for authentication and validation purposes, as well as mobile-friendly websites for the tutors so they can continue being mobile while working at the helpdesk.

The intended user group of the system are tutors and students working and seeking assistance at the Programming Helpdesk.

2.2 Objectives

1. Doubtfire should be extended to provide a in which to manage open and closed tickets created by students
2. Doubtfire should be extended to provide a way to inform all users at the helpdesk information regarding the current status of the queue (such as a projection onto a whiteboard with the current queue)
3. Doubtfire should be extended to collect analytics of the use of the Ticketing System, as well as open and closed tickets
4. A mobile-friendly web app should be developed for tutors while working at the helpdesk so that they can have access to their tickets on the go

2.3 Scope

- This system will only be used at Swinburne with direct access to the current Doubtfire system.
- The system will only be used by tutors at Swinburne, with direct support from the development team if needed.
- The system will be developed alongside current development of the Doubtfire project in order to support any API needs.
- The system will be developed and supported on Swinburne provided infrastructure such as servers and devices.

2.4 Critical Success Factors

CSF1: A basic form of the system is to be rolled out to the helpdesk for use at the completion of the project.

Basic here is referring to the 'barebones' ticketing system, that is, using the new system:

- Can students continue to receive assistance at the helpdesk?
- Can tutors continue to manage and supply assistance at the helpdesk?

CSF2: Have the system running on the existing Doubtfire infrastructure, the Rails server. If this basic form of the system cannot be rolled out at the completion of the unit, the project will be deemed a failure.

CSF3: The system can collect and present analytical data to the unit conveners regarding use of the Ticketing System and the Helpdesk.

2.5 Acceptance Criteria

Upon delivery, an acceptable product will need to demonstrate the ability to perform the following functional tasks:

2.5.1 Convenors

A convenor of a unit that employs Doubtfire within the unit that they convene will be able to use the system to view analytics related to the ticketing system being used at the helpdesk. Such analytics relate to:

- Number of tickets being opened
- Number of tickets being opened by each student
- Number of tickets being resolved
- Number of tickets being resolved by each member of teaching staff working at the helpdesk
- Number of tickets opened related to a unit
- Number of tickets opened related to a specific task within a unit
- On-peak and off-peak times for when tickets are opened
- Times within the academic semester (each day and each week) when the helpdesk is the busiest and quietest

2.5.2 Students

Student's that attend the helpdesk should be able to perform the following:

- A student will be able to sign onto the Programming Help Desk upon arrival

- A student will be able to open support tickets, where each ticket they open is related to a specific task within a specific unit that employs Doubtfire as a learning management system
- A student will be able to close a ticket that they have opened

2.5.3 Tutor

Tutors employed at the helpdesk should be able to perform the following:

- A tutor will be able to sign onto the Programming Help Desk upon the beginning of their shift
- A tutor will be able to use the ticketing system to view current support tickets that have been opened by students
- A tutor will be able to close an open ticket:
  (a) when the issue associated with the ticket has been resolved and
  (b) when a ticket has been opened and should not have been

3 Establishment

3.1 Process, Procedures and Standards

This is outlined in further detail under the Agile Workflow Documentation[2] and Coding Standards[3] documentation.

3.2 Project Environment

The physical environment of the Doubtfire Ticketing System will be at the Helpdesk (ATC620 at Swinburne University, Hawthorn Campus). The system operates in a room with desktop computers on each wall, running Windows 7, and several floating tables in the middle of the room for use with laptops.

The hardware in this room is maintained and updated by Swinburne ITS. User accounts are managed by ITS and authentication is handled by the SIMS system.

[2]See `https://github.com/final-year-project/documentation/wiki/S2-SDLC-Plan`
[3]See `https://github.com/doubtfire-lms/doubtfire-web/blob/develop/CONTRIBUTING.md`

The server in which Doubtfire is run on (`doubtfire.ict.swin.edu.au`) is hosted, maintained and run by Swinburne ITS. Swinburne ITS also maintain the MySQL RDBMS which stores confidential student information, as well as the Rails infrastructure needed by Doubtfire.

Using the system requires:

- A modern desktop or mobile web browser, kept up to date. Doubtfire **does not support** Internet Explorer 9 or below, and is intended to be used by up-to-browsers such as Safari, Chrome or Firefox.
- Any desktop or laptop running a modern operating system (Windows 10, OS X 10.10+, Linux distributions released within the last two years etc.)

3.3 Project Team Skills Requirements

Doubtfire is both a server-side and front-end web application, where the interface for the application is delivered through a web browser. Proficiencies and knowledge requirements are defined as thus:

3.3.1 API Server

Refer to the Doubtfire API README[4] for more information.

- **Ruby**[5]: The coding language which the server is developed in
- **Ruby on Rails**[6]: The server-side framework that allows Ruby code to run on a server
- **PostgreSQL**[7]: The object-relational database which is used during development

3.3.2 Web Interface

Refer to the Doubtfire Web README[8] for more information.

[4]See `https://github.com/doubtfire-lms/doubtfire-api`
[5]See `https://www.ruby-lang.org`
[6]See `http://rubyonrails.org`
[7]See `http://www.postgresql.org`
[8]See `https://github.com/doubtfire-lms/doubtfire-web`

- **JavaScript**[9] **and CoffeeScript**[10]: The coding language used to develop the front-end
- **SCSS**[11]: The styling syntax used to style Doubtfire
- **Bootstrap**[12]: A front-end framework used for styling
- **AngularJS 1.4**[13]: A front-end platform designed for building web applications

3.3.3 Additional knowledge

- **Git**[14]: A code versioning system imperative to development with regards to tracking code changes and developer contributions
- **GitHub**[15]: A website for hosting code repositories using Git tracking. Enables forking and pull-requests to merge code into the official Doubtfire codebase[16].
- **RESTful architectures**[17]: Representational state transfer architecture styling which most modern web-applications are based on.
- **Socket.IO**[18]: Socket-based JavaScript library used to build realtime web applications.

4 Activities, Deliverables and Capital Resources

4.1 Deliverables

A functioning extension of Doubtfire which will enable students and tutors to create and manage tickets. It will be completed using the Git development system currently outlined in the contributing documentation and merged into the live Doubtfire system by the client. This should include the analytics system.

[9] See https://www.javascript.com
[10] See http://coffeescript.org
[11] See http://sass-lang.com
[12] See http://getbootstrap.com
[13] See http://angularjs.org
[14] See https://git-scm.com
[15] See http://github.com
[16] See http://github.com/doubtfire-lms
[17] See https://en.wikipedia.org/wiki/Representational_state_transfer
[18] See http://socket.io

4.2 Activities and Tasks

There are a number of activities that the group members must embark on in order to ensure that the project's development and overall delivery a success. The key activities that have been identified as critical to the project's success are outlined and defined in detail below.

4.2.1 Research

Research is absolutely imperative in terms of understanding the need for the project and the domain for which the project is being developed for. Research, when done correctly, provides for a concrete foundation on which to build the project and to ensure that it is stable throughout the entire development cycle.

Without researching the project's target domain and the functionality that it will offer, the development would be based upon a loose construction of ideas that vaguely resemble what purpose the product is supposed to serve. Not only does research provide for a solid foundation on which to develop the project, it aids in developing better problem solving skills, critical thinking measures, confidence in what you're developing as well as project driven motivation.

4.2.2 Development

The development activity is where the knowledge gathered from the research phase is applied and constructed into a meaningful representation of what the project is supposed to represent. Throughout the development phase, each group member will be completely focussed on implementing a seamless solution.

4.2.3 Testing

A test-driven development (TDD) methodology will be adopted. This will include testing each and all of the different tool sets that, combined, produce the overall functionality of the ticketing system.

It is extremely important to conduct rigorous testing in order to ensure that the product is working exactly as is intended. It is also important to identify any issues that the system may have during the testing phase so that these may be corrected.

4.2.4 Documentation Authoring

After the project has been developed and tested rigorously and meets a standard of functionality that all of the group members can agree upon, documentation needs to be developed.

During this stage, user manuals and technical documentation will be authored in an interactive fashion (i.e., via a Git wiki) to provide for a complete dissection of the system and how it all works.

4.2.5 Submission for Critical Analysis

This task involves submitting the final project with all accompanying documentation for critical analysis from an individual(s) separate to the group dynamic. It is the intention of the group to have any such individual walk away from analysing the project feeling completely satisfied in the final product and all of the documentation provided with it.

4.3 Resources

4.3.1 Organisation Structure

The organisation structure of the helpdesk is listed in hierachy as thus:

1. **Helpdesk Administrator**, Andrew Cain
2. **Helpdesk Subject Convenors**:

- Andrew Cain, Introduction to Programming
- Alan Colman (`acolman@swin.edu.au`), Creating Web Applications
- Chris McCarthy (`cdmccarthy@swin.edu.au`), Object-Oriented Programming

3. **Helpdesk Subject Tutors**
4. **Helpdesk General Support Staff**
5. **Helpdesk Volunteers**

4.3.2 Development Equipment

UNIX-based systems to develop the system is required as Doubtfire does not support development on Windows.

5 Risk Analysis

5.1 Risk 1

- **Possible Issue:** Misunderstanding or poor interpretation either via electronic or verbal communication
- **Chance of Occurrence:** High
- **Severity of Detriment to Group Ambitions:** Little to none
- **Preventative Methods:**

1. Ask clear and concise questions
2. Communicate any ideas as soon as they dawn
3. Document clear notes
4. In the event that something is not clearly communicated, make the issue known

5.2 Risk 2

- **Possible Issue:** The group members all have different opinions in regards to how a particular problem should be approached
- **Chance of Occurrence:** High
- **Severity of Detriment to Group Ambitions:** Low to Medium
- **Preventative Methods:**

1. Elect to resolve the opinion dispute by casting a majority vote
2. Consult the project supervisor for their opinion

5.3 Risk 3

- **Possible Issue:** Task delegation becomes an issue because no group member elects to take on responsibility
- **Chance of Occurrence:** Medium
- **Severity of Detriment to Group Ambitions:** Low to Medium
- **Preventative Methods:**

1. Refer to the elected responsibilities
2. Consult project supervisor for indisputable delegation authority

5.4 Risk 4

- **Possible Issue:** Absent from Lectures
- **Chance of Occurrence:** Medium
- **Severity of Detriment to Group Ambitions:** Medium to High
- **Preventative Methods:**

1. Always inform group members if your attendance is in jeopardy
2. Understand that you're an integral part to the group's success

5.5 Risk 5

- **Possible Issue:** Absent from arranged group meetings
- **Chance of Occurrence:** Low
- **Severity of Detriment to Group Ambitions:** Medium to High
- **Preventative Methods:**

1. Always inform group members if your attendance is in jeopardy
2. Understand that you're an integral part to the group's success
3. Clearly communicate any commitment issues with fellow group members

5.6 Risk 6

- **Possible Issue:** Development timelines ignored/not met
- **Chance of Occurrence:** Low
- **Severity of Detriment to Group Ambitions:** Severe
- **Preventative Methods:**

1. Understand that the group is relying on a strong work ethic to produce exceptional results
2. Constantly check your delegated tasks and review deadlines
3. Understand your role within the group
4. Understand that without submissions, the whole group suffers

5.7 Risk 7

- **Possible Issue:** Document availability issues

- **Chance of Occurrence:** Low
- **Severity of Detriment to Group Ambitions:** Severe
- **Preventative Methods:**

1. Utilise a single medium for document sharing and collaboration
2. Ask questions and know which tasks are delegated to each group member

5.8 Risk 8

- **Possible Issue:** Transportation issues
- **Chance of Occurrence:** Low to Medium
- **Severity of Detriment to Group Ambitions:** Low
- **Preventative Methods:**

1. Use punctuality to ensure that your transport is not jeopardised
2. In the event that transport is absent, communicate your situation with fellow group members

5.9 Risk 9

- **Possible Issue:** Delegated task not completed
- **Chance of Occurrence:** Low
- **Severity of Detriment to Group Ambitions:** Severe
- **Preventative Methods:**

1. Offer a group communication with the group member who has not produced material.
2. Involve the subject convenor and arrange a remedy.

6 Schedule

6.1 Overview and Timeline

An overview of the semesterly schedules are outlined in Figure 1. The team plans to adopt a fortnightly sprint, surrounded by a week of sprint planning prior to the sprint beginning and a week of a sprint retrospective. Further information regarding the workflow is outlined in the Agile Workflow Summary[19].

[19]See `https://github.com/final-year-project/documentation/wiki/S2-SDLC-Plan`

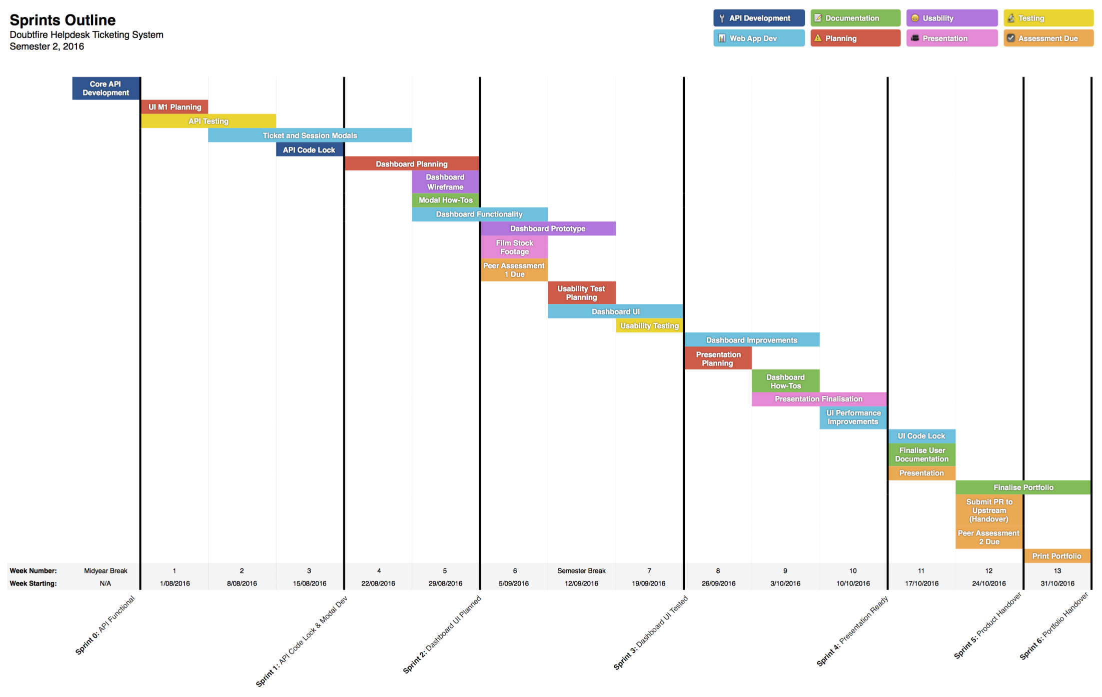

Figure 1: Sprint Plan for Semester 2

The overarching plan for deliverables will occur during a Sprint Retrospective. Within the Sprint Retrospective, work from the previous sprint is shown to the client, which serves as a feedback loop for the previous two-week iteration.

Thus, the general principle will be to:

1. Plan a sprint, deciding which tasks need to be allocated
2. Work on each task over a two-week sprint
3. Deliver new functionality to the client during the retrospective week
4. Improve upon the product given the client's feedback and work on this for the following week.

Whilst the dates are time-fixed, the durations of sprints, sprint planning and sprint retrospectives are *dynamic*, and so dates may not be adhered to concretely since they only act to serve a guide. For example, there may be a time where a two-week sprint is simply not enough time given the time allowed, and thus, the two-week schedule may be extended to a three or four week schedule, reducing planning and retrospective time.

This is chiefly due to external dependencies on the project, especially since the technical requirements will be forever changing with client demands.

6.2 External Dependencies

6.2.1 Client and Helpdesk Manager

As Andrew Cain is the lead developer for Doubtfire, and he is also the manager of the Programming Help Desk, his decisions, choices and influences largely sway the direction of the project.

The team will be working closely with Andrew to ensure that he and the team always stay on the same page in terms of the project. In an agile context, the team will conduct weekly 'catch up' meetings which act, serving as the stand-ups the team involves the client in.

6.2.2 Other Studies

Other final year subjects have large amounts of work during assessment times, which will sometimes overlap with development of the project.

6.2.3 Personal Life

This includes general physical and mental health of each team member, which is obviously more important than any work.

6.3 Assumptions Made

Completion of the project assumes that the relevant Swinburne services will aid us. ITS is to provide software and hardware required to run the system, and will also provide developmental resources as outlined in the previous sections. It is also expected that Swinburne will aid the team by providing the team with some suitable development room, as well as development hardware.

Another assumption made is that the helpdesk will still operate 'as-is'; the need for the project is still relevant throughout the year. There will be no unexpected downtime of the helpdesk or indeed the University.

7 Budget

The breakdown of hours have based on worklog hours. Both the semesterly estimates, and actual recordings are for Semester 2 are given. The sprint hours allocated for this semester have been indicated also.

Estimates for the project have been calculated as thus:

1. 96 hours minimum allocated for project work, either performed individually or together
2. 24 hours minimum allocated for team and client meetings
3. 12 hours minimum allocated for supervisor meetings

This gives a total of 132 hours per semester.

Team Member	S1 Estimation	S1 Actual	S2 Estimation	S2 Actual
Alex	132	135	132	XXX
Jake	132	146	132	XXX
Reuben	132	TBA	132	XXX
Lachlan	132	70.5	132	XXX

Team Member	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5
Alex	22	33	39	17	8
Jake	20	29	44	34	8
Reuben	15.5	34.75	TBA	TBA	TBA
Lachlan	14.5	18.8	24.7	5	5.5