

Doubtfire Helpdesk Ticketing System

SWE40002–Software Engineering Project B

SWINBURNE UNIVERSITY OF TECHNOLOGY

Team 20

Note to assessor(s): Every effort was made to convert our interactive Wiki at <http://github.com/final-year-project/portfolio/wiki> in this document. This was made possible by converting the pages to L^AT_EX using Pandoc. Should there be formatting issues, please refer to the original content on the GitHub wiki instead.

Declaration

We declare that this portfolio is a group assessment and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for us by another person.

We declare that all work in this portfolio has been sighted, reviewed and agreed by each team member, including a peer-validated review of worklog entries by each of us.

— October, 2016



Alex Cummaudo



Jake Renzella



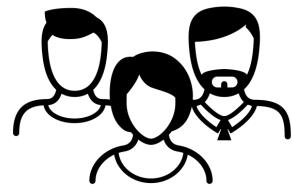
Lachlan West



Reuben Wilson

Contents

- Chapter 1 Group Details
- Chapter 2 Checklist
- Chapter 3 Assessment Criteria Agreement
- Chapter 4 Project Plan Update
- Chapter 5 Agile Methodology Workflow Update
- Chapter 6 Requirements Update
- Chapter 7 Design Prototype
- Chapter 8 Usability Testing Report
- Chapter 9 Technical Manual
- Chapter 10 User Manual
- Chapter 11 Source Code Directory Dump
- Chapter 12 Source Code
- Chapter 13 Meeting Minutes



Chapter 1

Group Details

Contents

1 Alex Cummaudo	2
2 Jake Renzella	2
3 Lachlan West	3
4 Reuben Wilson	3

1 Alex Cummaudo

Student Number: 1744070
Email: acummaudo@swin.edu.au
GitHub: alexcu
Roles:

- Team Lead
- Primary Developer
- UX/UI Developer
- Documentation



2 Jake Renzella

Student Number: 9993851
Email: jrenzella@swin.edu.au
GitHub: jakerenzella
Roles:

- Testing Co-Lead
- Project Developer
- UX/UI Developer
- Documentation
- Filming & Media



3 Lachlan West

Student Number: 2423480
Email: 2423480@student.swin.edu.au
GitHub: lachlanwest
Roles:

- Project Developer
- Documentation

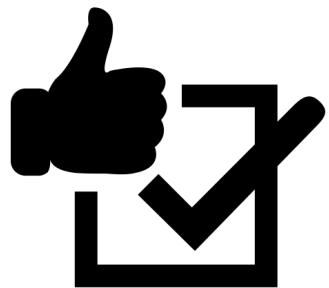


4 Reuben Wilson

Student Number: 9988289
Email: rwilson@swin.edu.au
GitHub: Reubsinit
Roles:

- Testing Co-Lead
- Project Developer
- UX/UI Developer
- Documentation
- Filming & Media





Chapter 2

Checklist

1 Portfolio Checklist

- Cover Sheet
- Checklist
- Research Report¹
- Updated² Project Plan
- Updated³ SDLC Plan
- Updated⁴ Requirements Specification Document
- UI Prototypes
- Usability Testing Report
- Technical Manual
- User Manual
- Table of Contents of Lab Manual⁵
- Source Code Directory Dump
- API + Web Source Code Changes
- MoU⁶
- CD - Source Code
- CD - Presentation

Notes:

1. Research Report was deprecated - instead of researching the topic, Andrew Cain opted for us to learn the foundations of how Doubtfire works by submitting small changes/bug-fixes. Evidence for this can be seen in the pull requests made from the Doubtfire code base by each team member during Semester 1 and midyear break:
 - Web Changes ([https://github.com/doubtfire-lms/doubtfire-web/graphs/contributors?
from=2016-03-01&to=2016-08-01&type=c](https://github.com/doubtfire-lms/doubtfire-web/graphs/contributors?from=2016-03-01&to=2016-08-01&type=c))
 - API Changes ([https://github.com/doubtfire-lms/doubtfire-api/graphs/contributors?
from=2016-03-01&to=2016-08-01&type=c](https://github.com/doubtfire-lms/doubtfire-api/graphs/contributors?from=2016-03-01&to=2016-08-01&type=c))
2. Project Plan Updated as per requirements changes made in mid-year break
3. Ditto
4. Ditto
5. Lab Manual not applicable
6. MoU not applicable to Open Source software. Refer to Doubtfire's License (<https://github.com/doubtfire-lms/doubtfire-api/blob/develop/LICENSE>).



Chapter 3

Assessment Criteria Agreement

Contents

1	Intended Learning Outcomes	2
2	Group Member Roles	2
2.1	Alex Cummaudo	2
2.2	Jake Renzella	3
2.3	Lachlan West	3
2.4	Reuben Wilson	3
3	Assessment Requirements	3
3.1	Required Documents	3
3.1.1	Work Log	3
3.1.2	Contribution Statement	4
3.1.3	Peer Review Documentation	4
3.1.4	Self Assessment Report	4
3.2	Grade Breakdown	4
3.2.1	Pass	5
3.2.2	Credit	5
3.2.3	Distinction	5
3.2.4	High Distinction	6

1 Intended Learning Outcomes

Our team's assessment criteria agreement is based upon a subjective analysis of work contribution to the project that relates to the Intended Learning Outcomes of the Final Year Project unit. For reference, they are as thus:

1. Apply professional practice, including active and consistent participation, delivery of technical presentations, reflection, and adherence to ethical codes of conduct as a member of a software development team
2. Apply software engineering methods and contemporary software development tools to the scoping, analysis, and design of a software system to meet client needs
3. Communicate proficiently with project stakeholders, and function as an effective member or leader of a development team in project scoping, analysis and design activities
4. Conduct a critical analysis and evaluation of aspects relevant to a software development project and justify implications for project directions

Each of these learning outcomes are to be demonstrated in the Required Documents as a part of an individual group member's submission.

2 Group Member Roles

Each group member has a self-associated role that indicates their primary area of contribution to the project.

2.1 Alex Cummaudo

Alex will focus in the following key areas, in order of importance:

1. Team Lead
2. Primary Developer
3. UX/UI Developer
4. Documentation

2.2 Jake Renzella

Jake will focus in the following key areas, in order of importance:

1. Testing Co-Lead
2. Project Developer
3. UX/UI Developer
4. Documentation
5. Filming & Media

2.3 Lachlan West

Lachlan will focus in the following key areas, in order of importance:

1. Project Developer
2. Documentation

Lachlan will develop these roles over time, and therefore his area of expertise may develop into new areas not yet listed.

2.4 Reuben Wilson

1. Testing Co-Lead
2. Project Developer
3. Research
4. Filming & Media

3 Assessment Requirements

3.1 Required Documents

3.1.1 Work Log

Each group member must produce a work log, which clearly outlines how they have allocated and spent time on the project. Work logs outline any such activity that is related to bringing the project to a state of completion.

3.1.2 Contribution Statement

Each group member must produce a project contribution statement, endorsed by each of the other group members. The contribution statement should outline every aspect of contribution toward the project made by the group member that the contribution statement belongs to.

It is the responsibility of the group to collectively decide what grade bracket each contribution statement is worth in a meeting at the end of semester.

3.1.3 Peer Review Documentation

Each group member must produce relevant peer review documents¹ for each member of the group, including themselves. Peer review provides for the ability for each group member to rate each group member against a number of key metrics.

3.1.4 Self Assessment Report

Each group member must produce a Self Assessment Report, outlining how their contributions to the overall goal of the group align to the Intended Learning Outcomes. Furthermore, each group member must use the Self Assessment Report in order to **justify the grade that they have specified that they deserve based on their contributions to the project.**

It is **compulsory** that the Self Assessment Report clearly outlines all aspects of individual contribution to the project.

3.2 Grade Breakdown

Grade breakdowns are outlined below. In order to be eligible for grading, each group member must provide all required documents, as outlined above, as well as meeting all of the criteria specified within the target grade bracket.

¹See <https://github.com/final-year-project/documentation/wiki/Deliverables#7-peer-reviews>

3.2.1 Pass

Any group member who has done **the bare minimum** in regards to overall project contribution and progress. Such a grade would be awarded to a group member who has:

- **little to no input** in regards to project documentation, and
- **expressed no interest in voicing ideas** about actual project design and development requirements

Minimum amounts of actual project source code contribution would also contribute to such a grade being awarded to a group member.

3.2.2 Credit

Any group member who has produced **satisfactory amounts of work** in regards to project contribution.

Such a grade would be awarded to a group member who has:

- **little to moderate input** in regards to project documentation, and
- **has expressed some interest in voicing ideas** about actual project design and development requirements.

Any such group member who has shown reasonable initiative in regards to contributing to actual project source code would demonstrate eligibility for a credit.

3.2.3 Distinction

Any group member who has produced **significant amounts of work** in regards to project contribution.

Such a grade would be awarded to a group member who has:

- **moderate to large amounts of input** in regards to project documentation, and
- **has expressed significant interest in voicing ideas** about actual project design and development requirements.

Any such group member who has shown large amounts of initiative in regards to contributing to actual project source code would demonstrate eligibility for a distinction.

3.2.4 High Distinction

Any group member who has produced **significant amounts of work, showing initiative in new areas**, in regards to project contribution.

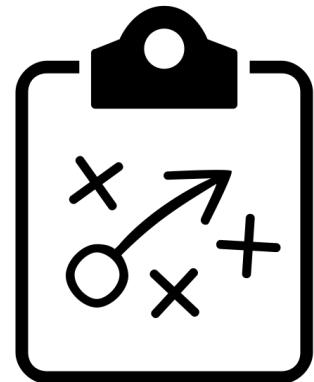
Such a grade would be awarded to a group member who has:

- **significant amounts of input** in regards to project documentation, and
- has **expressed significant interest in voicing ideas** about actual project design and development requirements.

Any such group member who has shown **large amounts of initiative** in regards to contributing to actual project source code would demonstrate eligibility for a distinction.

Additionally, any such group member who takes ownership over a particular aspect of the project development and demonstrates leadership in that field would be eligible for a High Distinction.

For example, a group member may decide to take ownership of unit testing and by doing so, commits to producing any relevant documentations in order to outline their contribution to that specific field and in order to educate the rest of the group about how unit testing is imperative to project integrity and progress.



Chapter 4

Project Plan Update

Contents

1 Introduction	3
1.1 Purpose of this document	3
1.2 Background	3
1.2.1 Swinburne University Programming Helpdesk	3
1.2.2 Doubtfire Learning Management System	3
1.3 Key Project Personnel	4
1.3.1 Client	4
1.3.2 Stakeholders	4
1.3.3 Project Manager	4
1.3.4 Project Members	4
2 Terms of Reference	4
2.1 Goals	4
2.2 Objectives	5
2.3 Scope	5
2.4 Critical Success Factors	5
2.5 Acceptance Criteria	6
2.5.1 Convenors	6
2.5.2 Students	6
2.5.3 Tutor	7
3 Establishment	7
3.1 Process, Procedures and Standards	7
3.2 Project Environment	7
3.3 Project Team Skills Requirements	8
3.3.1 API Server	8
3.3.2 Web Interface	8
3.3.3 Additional knowledge	9
4 Activities, Deliverables and Capital Resources	9
4.1 Deliverables	9
4.2 Activities and Tasks	10
4.2.1 Research	10

4.2.2	Development	10
4.2.3	Testing	10
4.2.4	Documentation Authoring	11
4.2.5	Submission for Critical Analysis	11
4.3	Resources	11
4.3.1	Organisation Structure	11
4.3.2	Development Equipment	11
5	Risk Analysis	12
5.1	Risk 1	12
5.2	Risk 2	12
5.3	Risk 3	13
5.4	Risk 4	13
5.5	Risk 5	13
5.6	Risk 6	14
5.7	Risk 7	14
5.8	Risk 8	14
5.9	Risk 9	15
6	Schedule	15
6.1	Overview and Timeline	15
6.2	External Dependencies	17
6.2.1	Client and Helpdesk Manager	17
6.2.2	Other Studies	17
6.2.3	Personal Life	17
6.3	Assumptions Made	17
7	Budget	18

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 Details¹ 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.

¹See <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 way 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 Documentation² and Coding Standards³ 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.

²See <https://github.com/final-year-project/documentation/wiki/S2-SDLC-Plan>

³See <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 README⁴ 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 README⁸ for more information.

⁴See <https://github.com/doubtfire-lms/doubtfire-api>

⁵See <https://www.ruby-lang.org>

⁶See <http://rubyonrails.org>

⁷See <http://www.postgresql.org>

⁸See <https://github.com/doubtfire-lms/doubtfire-web>

- **JavaScript⁹** and **CoffeeScript¹⁰**: The coding language used to develop the front-end
- **SCSS¹¹**: The styling syntax used to style Doubtfire
- **Bootstrap¹²**: A front-end framework used for styling
- **AngularJS 1.4¹³**: A front-end platform designed for building web applications

3.3.3 Additional knowledge

- **Git¹⁴**: A code versioning system imperative to development with regards to tracking code changes and developer contributions
- **GitHub¹⁵**: A website for hosting code repositories using Git tracking. Enables forking and pull-requests to merge code into the official Doubtfire codebase¹⁶.
- **RESTful architectures¹⁷**: Representational state transfer architecture styling which most modern web-applications are based on.
- **Socket.IO¹⁸**: 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.

⁹See <https://www.javascript.com>

¹⁰See <http://coffeescript.org>

¹¹See <http://sass-lang.com>

¹²See <http://getbootstrap.com>

¹³See <http://angularjs.org>

¹⁴See <https://git-scm.com>

¹⁵See <http://github.com>

¹⁶See <http://github.com/doubtfire-lms>

¹⁷See https://en.wikipedia.org/wiki/Representational_state_transfer

¹⁸See <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 hierarchy 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.

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 Summary¹⁹.

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.

¹⁹See <https://github.com/final-year-project/documentation/wiki/S2-SDLC-Plan>

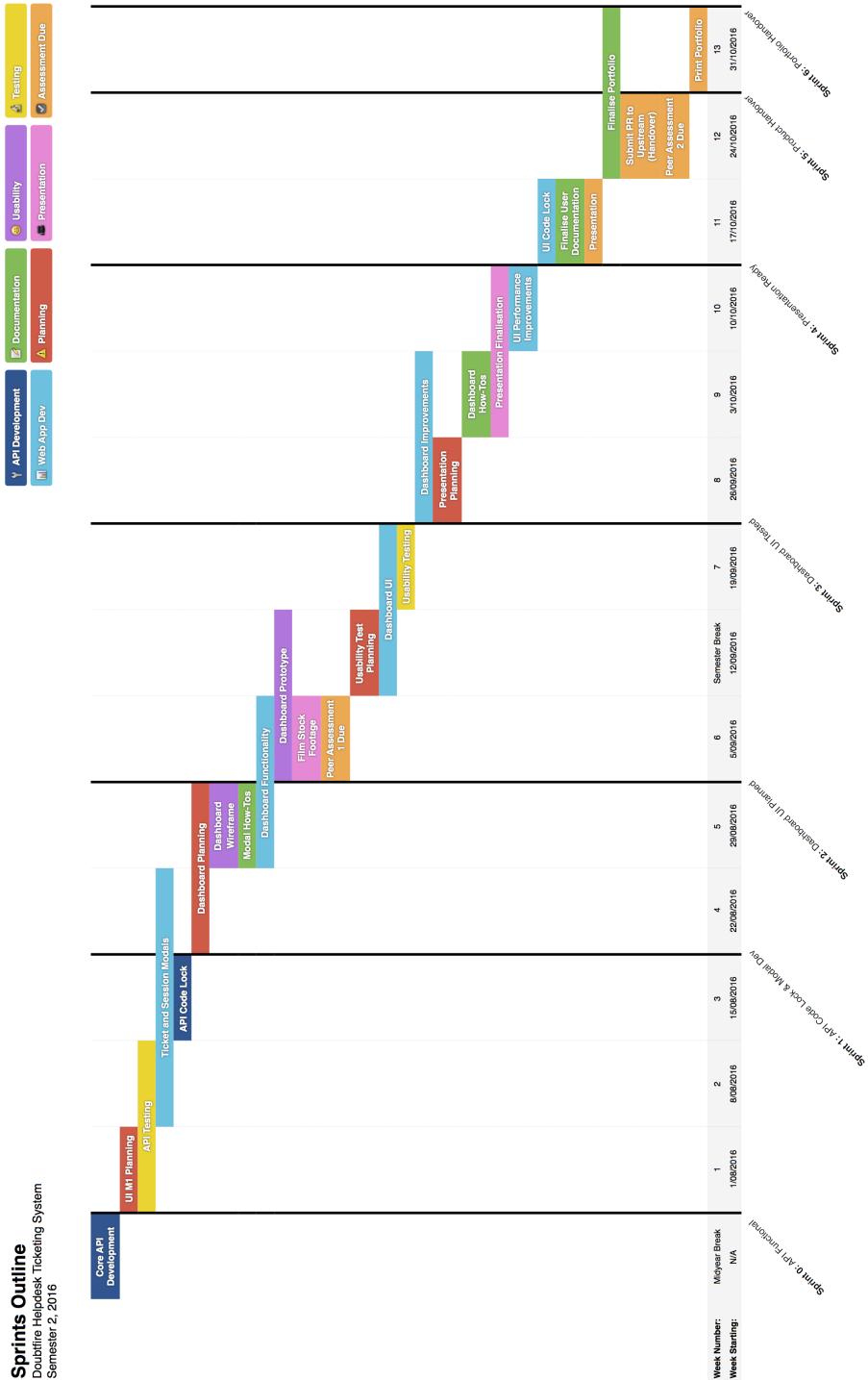


Figure 1: Sprint Plan for Semester 2

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.

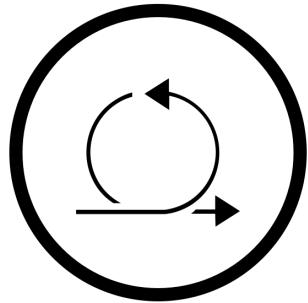
Estimates for the project have been calculated as thus:

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
Alex	132	135	132
Jake	132	146	132
Reuben	132	TBA	132
Lachlan	132	70.5	132

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



Chapter 5

Agile Methodology Workflow Update

SDLC Plan

Contents

1 Development Framework	2
2 Sprints	2
2.1 Sprint Dates & Goals	2
2.1.1 Semester 1	2
2.1.2 Midyear Break	3
2.1.3 Semester 2	3
3 Development workflow	4
3.1 High level overview	4
3.2 Trello Boards	5
3.2.1 Doubtfire Backlog Board	5
3.2.2 Helpdesk Ticketing System Board	6
3.2.3 Trello Workflow in a Sprint	8

1 Development Framework

The team has decided to adopt a **Scrum framework** in our **Agile approach** to deliver this software.

2 Sprints

The team will adopt multiple sprints throughout the duration of the project, whose dates and goals are defined by a group decision and with the client.

2.1 Sprint Dates & Goals

The following lists each of the Sprints, their dates and intended goals. This list will be updated as new sprints are devised throughout the development lifecycle.

2.1.1 Semester 1

1. **March 7 to March 21** - This sprint was decided in the March 7 meeting¹ with our client and is intended to “consist of small bug fixes or enhancements that will help the team familiarise themselves with Doubtfire’s codebase”
2. **March 21 to April 4** - This sprint builds upon the first sprint whilst concurrently working on requirements documents and assessment criteria for the project. Jake and Reuben will take on unit testing as it has been neglected in the past for the Rails API and will investigate methods to test it.
3. **April 4 to May 4** - This sprint is the largest sprint of the semester. It focuses on the team dedicating time to shift their resources on a majority of requirements analysis for the system, including prototyping and architecture decisions. Like Sprint 2, it is expected that team members also work on tasks as need be for practise and familiarisation of the Doubtfire codebase
4. **May 4 to May 23** - This sprint focuses on the film which needs to be shot by the presentation on May 23 in Week 11. In addition, this sprint will focus on the majority of the requirements documentation that needs to be properly thought about, such as use case descriptions.

¹See <https://github.com/final-year-project/documentation/wiki/Meeting-Minutes#re-4-spiking-doubtfire-for-the-next-few-weeks>

5. **May 23 to May 30** - This last sprint for Semester 1 wraps up the first semester as the team prepares their portfolio for submission and assessment. It should include a thorough review of the documents produced thus far, and make additional ‘last-minute’ changes if need be.

2.1.2 Midyear Break

0. **May 30 to August 1** - This sprint, over the midyear break, aims to develop the core API codebase changes needed to introduce the helpdesk ticketing system. It is aliased to Semester 2’s ‘zero’th’ sprint.

2.1.3 Semester 2

1. **August 1 to August 22** - This first Semester 2 sprint aims on locking the API code. All testing for the API should be done by this stage. Begin planning the UI changes needed such as modals to create tickets and sessions which tutors can clock on/clock off from the programming helpdesk.
2. **August 22 to September 5** - This sprint aims at having the Dashboard UI planned in addition to beginning user manual documentation on how to create and submit tickets as well as implementing some partial functionality of the dashboard.
3. **September 5 to September 26** - This sprint aims at completing as much usability testing as possible on the dashboard, being the most integral part of the software.
4. **September 26 to October 10** - This sprint focuses on getting the dashboard finalised and adding in as many performance improvements to the UI as possible. The team also will focus on the presentation and getting it finalised in this time.
5. **October 10 to October 31** - This sprint will focus on finalising the codebase and resolving all conflicts between the origin Doubtfire codebase and the new Doubtfire codebase with new features added.
6. **October 31 to November 6** - This sprint will focus on the team finalising the portfolio for printing and submission.

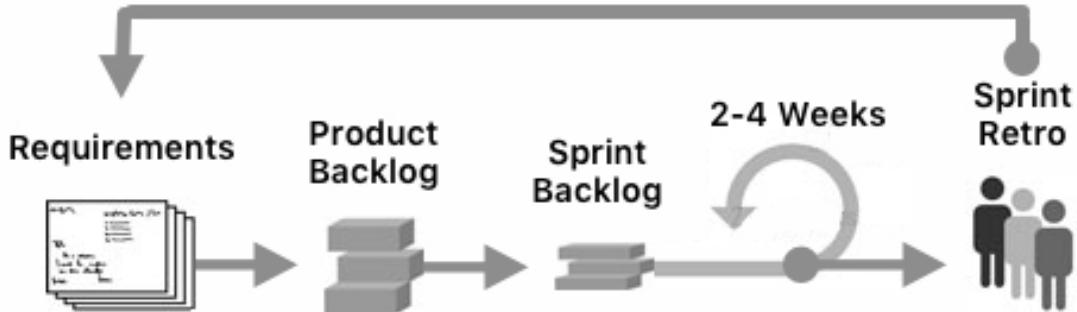


Figure 1: High level overview of the workflow

3 Development workflow

3.1 High level overview

The high-level overview of the workflow is an adapted standard scrum process that better suits the needs of the project:

1. The **requirements** of the project is devised with the client in **ongoing client meetings**. This means that requirements can be added on an ongoing basis throughout the project
2. Requirements are fed into the **project backlog**, that is the backlog of the entire Doubtfire system, as tasks
3. Specific tasks are chosen to be worked on a sprint into a **sprint backlog**
4. The sprint lasts for 2-4 weeks, where tasks from the sprint backlog are eventually all completed
5. A **sprint retrospective** is run amongst both the team and the client, where feedback from the sprint is eventually fed back into initial the requirements

There are no daily stand-ups like the standard scrum framework mandates. This is due to time and constraints with the group, as only one or two meetings are possible throughout the week. The team aims for a meeting together once a week, and a meeting with the client on a fortnightly basis.

To slimline the workflow, the sprint review process is merged into the sprint retrospective, meaning the retrospective is conducted with the team and client such that the client gets a feeling of how the team is progressing through tasks *and* can suggest changes to the product in one go.

The workflow enhances Doubtfire's **continuous integration**² as each changes will be pushed to the `develop` branch, and eventually `master` branch (meaning that as tasks get completed, the code changes will go live to production).

3.2 Trello Boards

As described by the Project Tools³ document, Trello is being used to manage tasks. The Trello Workflow adapts to the high level overview in the form of variant boards and columns. Each board and its intended workflow is outlined below in further detail.

3.2.1 Doubtfire Backlog Board

The **Doubtfire Backlog Board**⁴ is the *Product Backlog* for Doubtfire. It aims to outline all tasks in Doubtfire's backlog horizontally. Each column is described as thus:

1. **Fresh Ideas** - New tasks that the product owner thinks of will be added here. But those tasks should be moved into one of the other lists on this board as soon as possible, and therefore this list should be kept as empty as possible.
2. **Quick and Easy**
3. **Top Priority** - Tasks that need to be completed ASAP, such as critical bugs
4. **High Priority** - Tasks that need to be completed which have high importance, such as bugs
5. **Medium Priority** - Tasks that should be completed soon, such as new features or enhancements
6. **Low Priority** - Tasks that would be nice to have done eventually, such as small UI beautifications
7. **One Day** - Ideas that would be great to have *one day* if we had the time, essentially tasks that are currently too out of scope

²See https://en.wikipedia.org/wiki/Continuous_integration

³See <https://github.com/final-year-project/documentation/wiki/Project-Tools#trello-task-management>

⁴See <https://trello.com/b/0uh6AZdu/doubtfire-backlog>

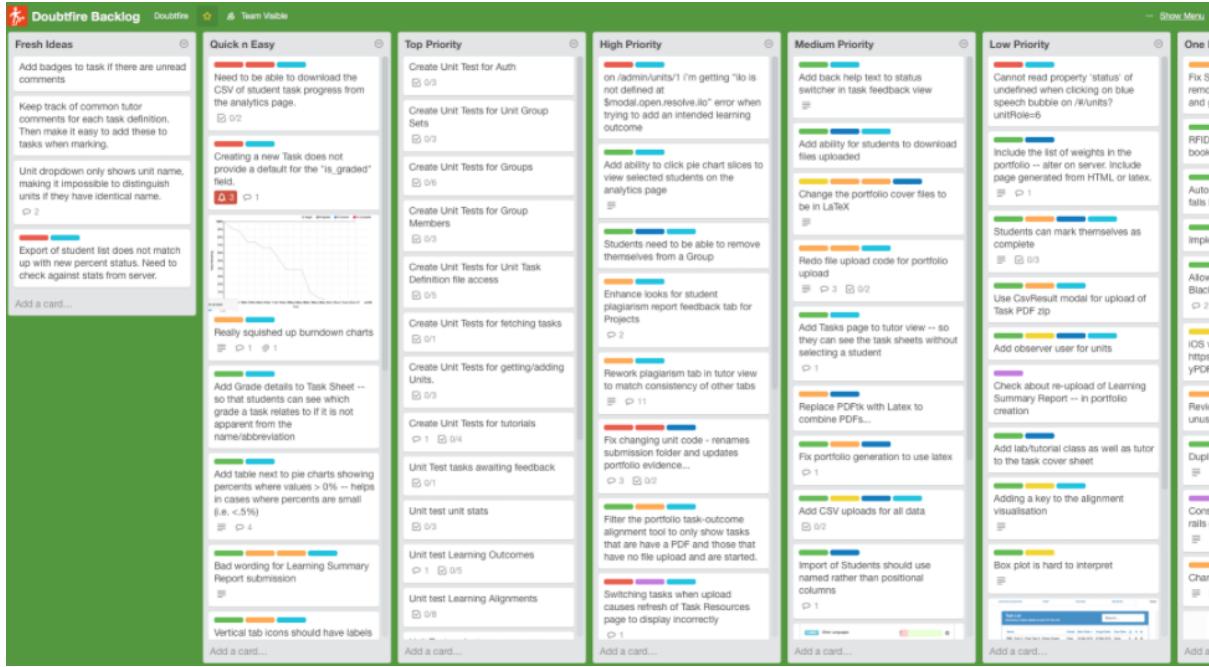


Figure 2: The Doubtfire Backlog Board

8. Maybe... - Ideas that are still being considered

The Doubtfire Backlog Board is maintained by the Product Owner (i.e., the client)—the team does not have access to add new tasks here unless it is approved by the Product Owner.

3.2.2 Helpdesk Ticketing System Board

The **Helpdesk Ticketing System Board**⁵ is the board that outlines all tasks for the team's project.

This board not only contains the **Sprint Backlog** tasks, but also meta tasks related to administration and assessment that needs to be done (such as organising meetings with clients, installing required software etc). This will help the project manager organise which tasks the team needs to not only on a *developmental* basis, but also on an *assessment* and *administration* basis for the final year project unit.

The board is organised into several columns:

⁵See <https://trello.com/b/8a1k0Wud/helpdesk-ticketing-system>

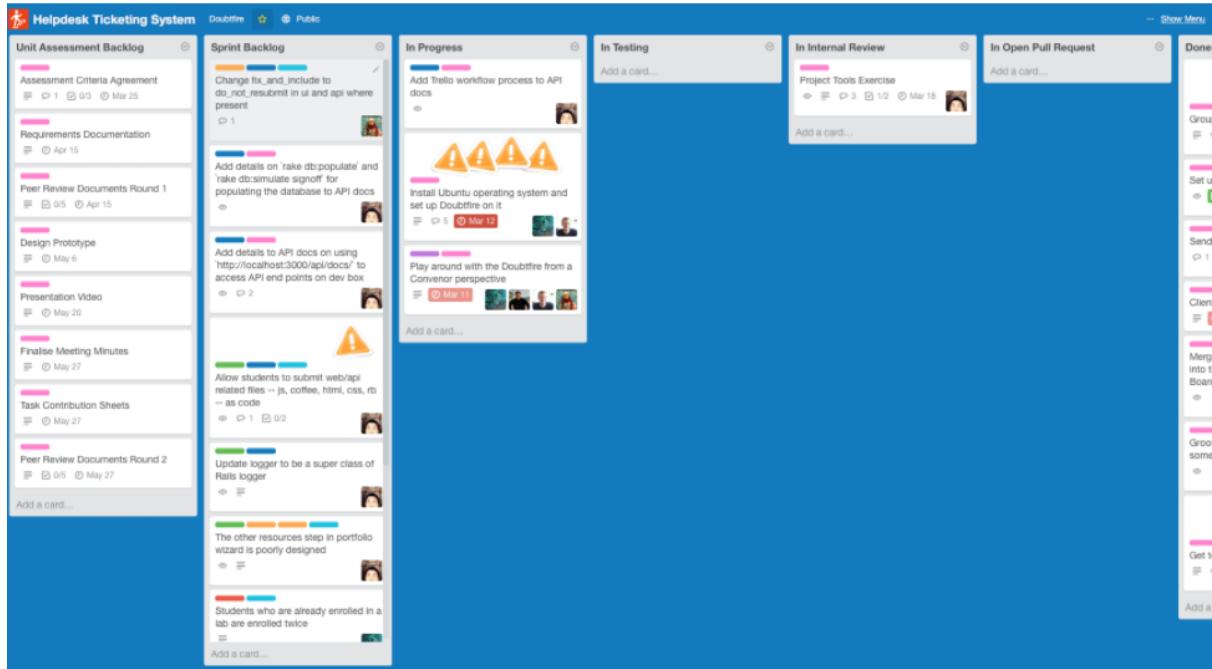


Figure 3: The Helpdesk Ticketing System Board

1. **Unit Assessment Backlog** - The backlog of tasks that are required by the final year project unit deliverables⁶
2. **Sprint Backlog** - Tasks involved over the upcoming time-fixed sprint moved from the Doubtfire Backlog board - essentially the current sprint backlog
3. **In Progress** - Tasks that are currently in progress. These tasks **must be assigned to whoever is working on them**
4. **In Testing** - Where relevant, tasks move into this column if unit or integration testing is needed on that task (e.g., a new API endpoint should have unit tests written). **These tasks should be assigned to whoever is writing the tests for the task**
5. **In Internal Review** - Tasks that are to be reviewed internally by another team member. **These tasks should be reassigned to the reviewer**
6. **In Open Pull Request** - Tasks that have been put into a Pull Request and assigned to a product owner for external code review. **These tasks should be reassigned to the external code reviewer.**

⁶See <https://github.com/final-year-project/documentation/wiki/Deliverables>

7. **Done** - When the task is merged into the `develop` branch (the Pull Request has been closed).

This board is maintained by the project manager of the team, and also updated by team members as they progress through their allocated tasks.

3.2.3 Trello Workflow in a Sprint

This workflow is used to guide team members on how to use Trello for their day-to-day activities whilst working on Doubtfire.

The workflow also complements the Doubtfire Git Workflow⁷, meaning that all changes made will be pushed into the primary `develop` branch on the Doubtfire product workflow for improved continuous integration.

3.2.3.1 Prepare cards for a sprint

In this step, team members will go to the Doubtfire Backlog Board⁸ and find (or create with the Product Owner's permission) tasks that are relevant for the current sprint.

Once they have found a card, they can move it directly to the Helpdesk Ticketing System Board⁹'s Sprint Backlog column. To do this, they can click on a card and then move it:

3.2.3.2 Progressing the card

Once the card is in the backlog on the Helpdesk Ticketing System Board, it will be progressed throughout the board (moved toward to the right hand side of the board) to describe its process by dragging and dropping it between columns:

3.2.3.2.1 In Progress

When the card is ready to start on, progress the card from the **Sprint Backlog** to the **In Progress** column.

It needs to be assigned to whoever is working on the card. To do so, click on the card and assign the task to a team member:

⁷See <https://github.com/doubtfire-lms/doubtfire-api/blob/develop/CONTRIBUTING.md>

⁸See <https://trello.com/b/0uh6AZdu/doubtfire-backlog>

⁹See <https://trello.com/b/8a1k0Wud/helpdesk-ticketing-system>

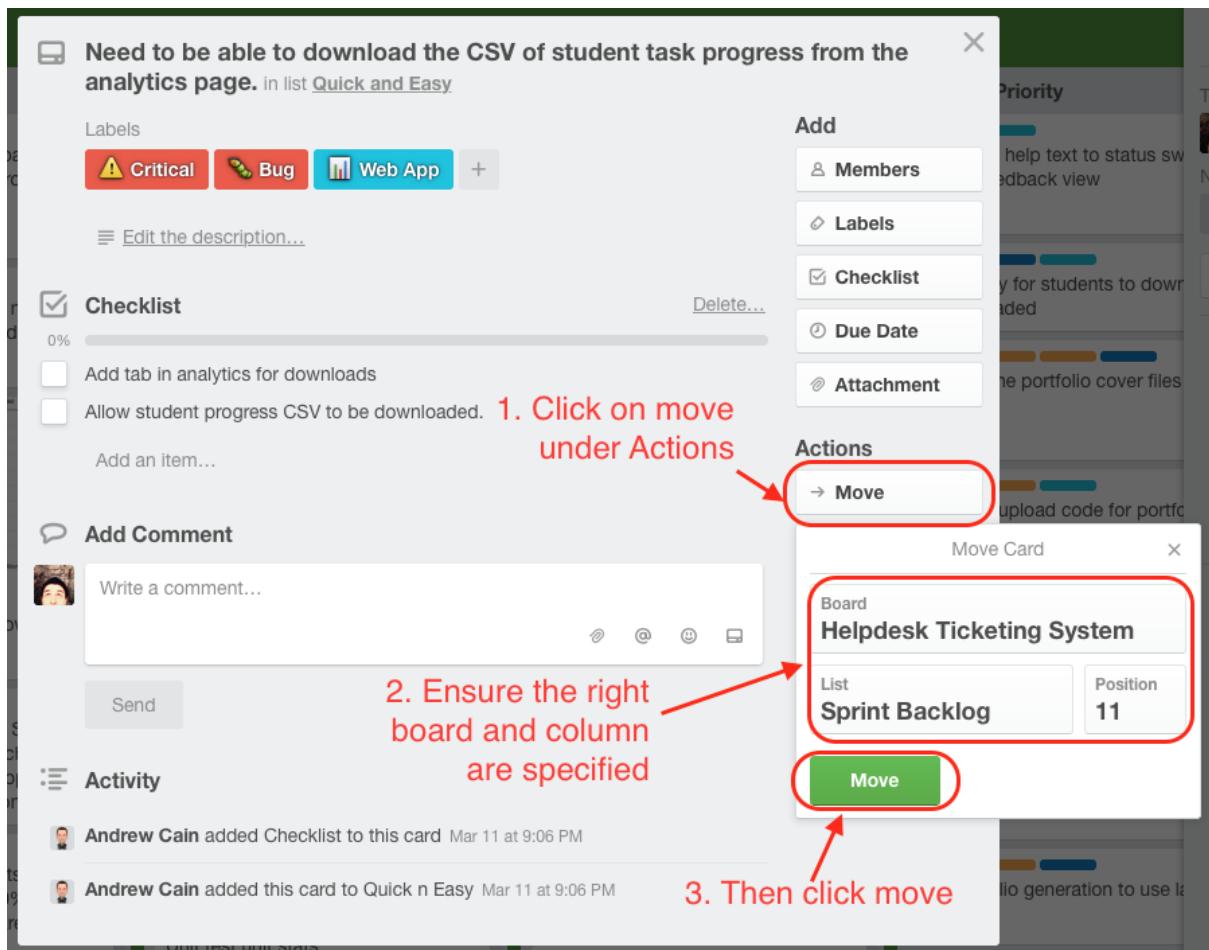


Figure 4: Moving a card to the right board

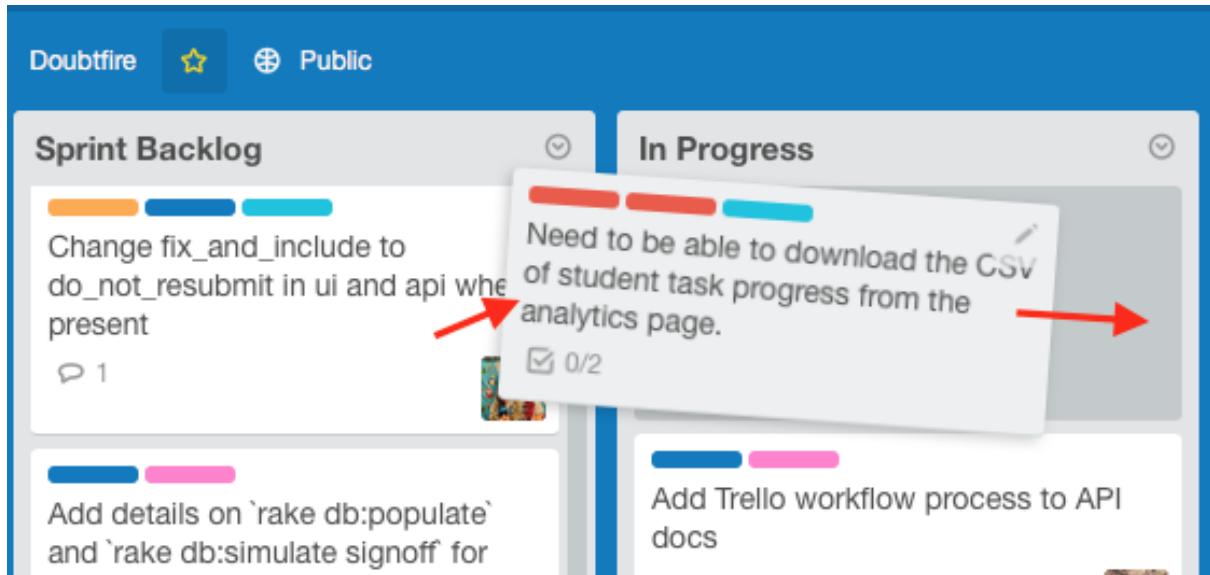


Figure 5: Dragging and dropping the card

3.2.3.2.2 In Testing

If the card requires testing to be done, progress the task from **In Progress** to **In Testing**.

3.2.3.2.3 In Internal Review

When the person working on the card thinks the card is ready, they should run a **code walkthrough** with someone else in the team.

Progress the task from **In Testing** or **In Progress** to **In Internal Review** and sit down together with the team member. Walk the other team member through both the *functionality* changes added, as well as the *code* that has been added. This ensures for optimal product quality and code quality by having a second set of eyes look over the code.

When the code review is over, remove the other team member from the members list of the card.

3.2.3.2.4 In Open Pull Request

As through the Doubtfire Git Workflow¹⁰ a Pull Request should be submitted to the

¹⁰See <https://github.com/doubtfire-lms/doubtfire-api/blob/develop/CONTRIBUTING.md#>

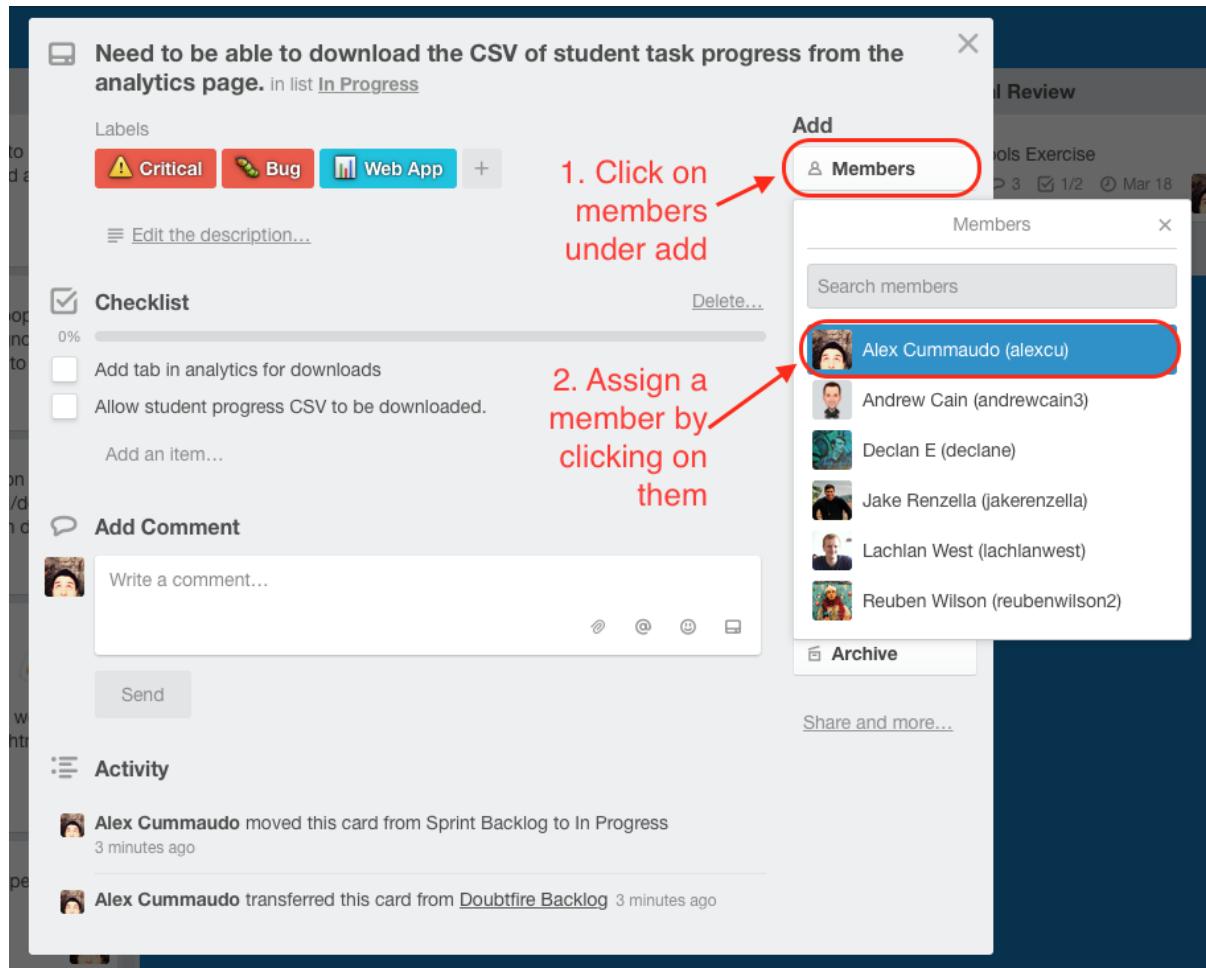


Figure 6: Assign a member

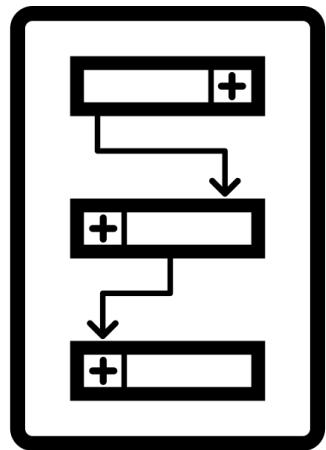
Product Owners (i.e., the `doubtfire-lms` repository) when ready for external review.

Progress the card to **In Open Pull Request** from **In Internal Review** when the pull request has been submitted and assign the Product Owner (e.g., Andrew Cain) to the card (if applicable).

3.2.3.2.5 Done

Progress the card from **In Open Pull Request** to **Done** only once the changes made from the card have been merged and the Pull Request is closed. The assigned task member can be removed from the card at this point.

⁴-submitting-a-pull-request-pr-to-the-upstream-repository



Chapter 6

Requirements Update

Contents

1 Goals & Objectives	2
2 Entities	2
3 Use Cases	3
3.1 Students	3
3.1.1 Submit a ticket	3
3.2 Tutors	4
3.2.1 Clocking On	4
3.2.2 Getting the next ticket off the ticket queue	4

1 Goals & Objectives

Whilst reading this documentation, it is important to keep the following goals and objectives in mind:

The Doubtfire Helpdesk Ticketing System will provide:

- A way to improve efficiency of helping students
- A way for tutors to track which students need help
- A way to manage tutor clock-on times
- A way for convenors to see how much their students utilise the helpdesk and at what times
- A way for convenors to see how their tutors are clocking on at the helpdesk

Refer to the requirements documentation for more on this.

2 Entities

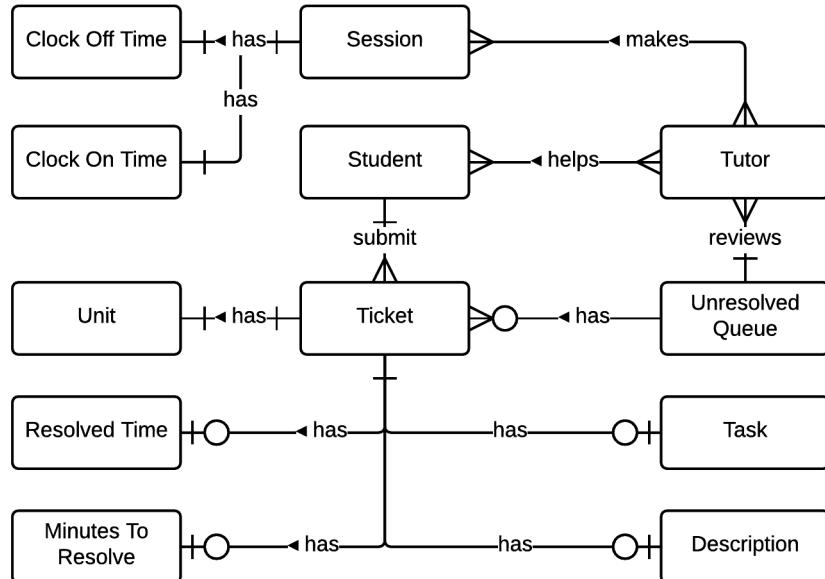


Figure 1: ERD

An **unresolved queue** is a list of *all unallocated tickets* that have been submitted at the helpdesk. Tutors working at the helpdesk aim to keep this global queue as minimal as

possible; when a new ticket is created by a student, tutor's may resolve or close tickets. Students may close their tickets should they not want further help.

3 Use Cases

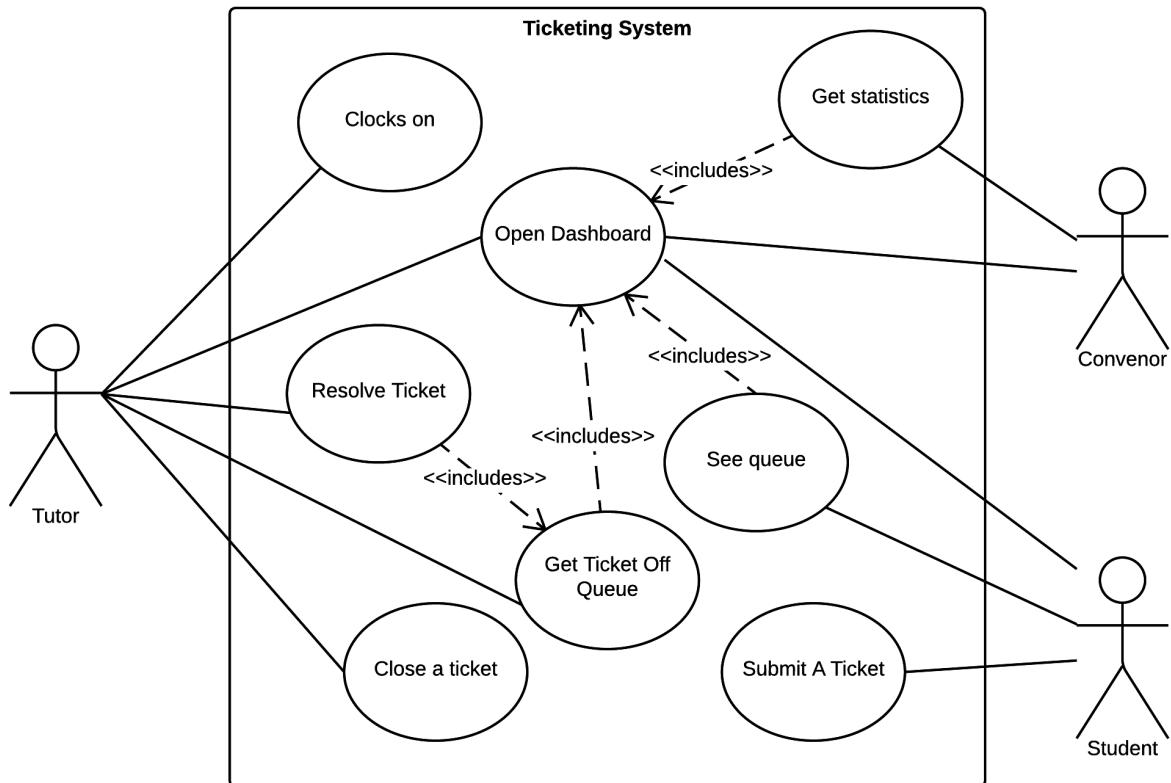


Figure 2: Use Case Diagram

3.1 Students

3.1.1 Submit a ticket

3.1.1.1 Primary Use Case

- Step 1. Student signs into Doubtfire
- Step 2. Student selects Helpdesk from header
- Step 3. Student selects unit they want help with
- Step 4. Student submits the ticket.
- Step 5. Student views an estimate of wait time

3.1.1.2 Alternate Use Cases

Student doesn't have a computer

- Step 1a.** Student goes to instructor PC **Step 1b.** Student enters in their student ID
Step 1c. Continue from (3)

Helpdesk ticket queue is overloaded

- Step 2a.** Student is given a visual notice that they might have to wait a while to get help **Step 2b.** Student *optionally* cancels the process **Step 2c.** Student *optionally* continues from (3)

3.2 Tutors

3.2.1 Clocking On

3.2.1.1 Primary Use Case

- Step 1.** Tutor selects clock on from Doubtfire menu **Step 2.** Tutor enters how long they will work for. **Step 3.** Doubtfire will automatically clock off tutor at that time.

3.2.1.2 Alternate Use Cases

Tutor has an emergency and must leave before automatic clock off time

- Step 3a.** Tutor opens the Helpdesk from Doubtfire menu and manually clocks off.

3.2.2 Getting the next ticket off the ticket queue

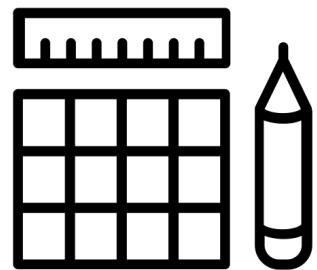
3.2.2.1 Primary Use Case

- Step 1.** Tutor taps the ticket of the student **Step 2.** Tutor attends and assists the student **Step 3.** Tutor marks the ticket as resolved

3.2.2.2 Alternate Use Case

Student isn't physically present

- Step 2a.** Tutor postpone's the ticket; goes onto the next ticket instead.



Chapter 7

Design Prototype

Contents

1 Dashboard Wireframe	2
1.1 Iteration One	2
1.2 Iteration Two	3
1.2.1 Iteration Three	3

1 Dashboard Wireframe

Each revision of the wireframes went through group discussion and iteration.

1.1 Iteration One

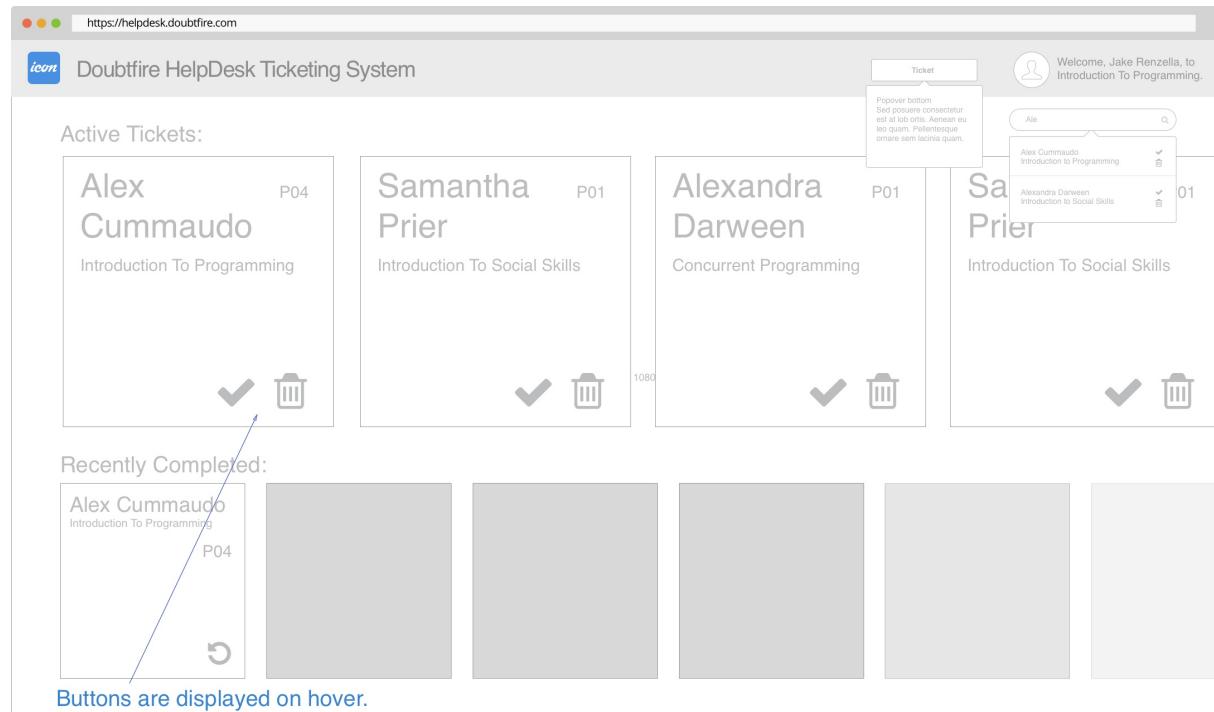
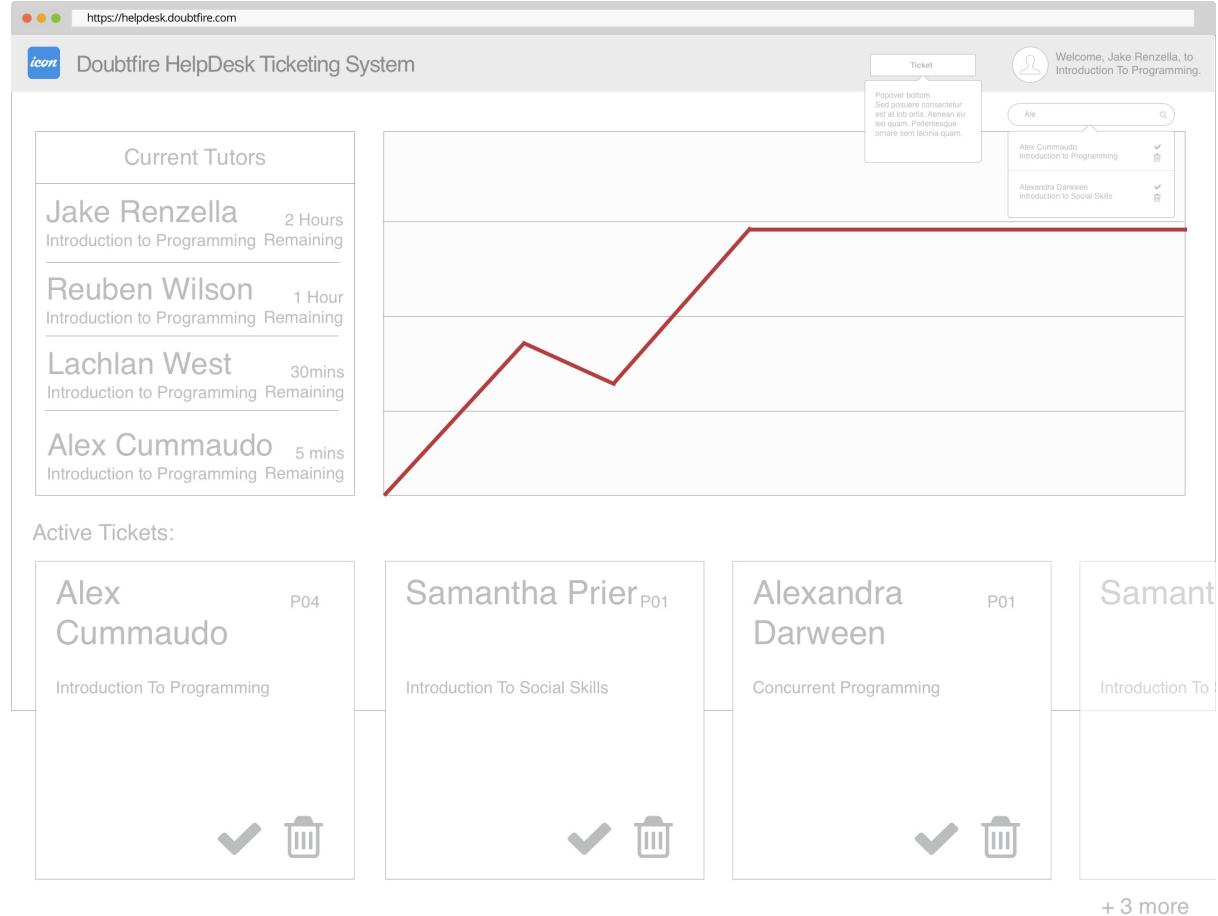


Figure 1: Iteration One of the Dashboard

Iteration one showed recently closed tickets, which was deemed unnecessary and a waste of space, so was removed in the next iterations. The general structure for the design was continued.

1.2 Iteration Two



The next iteration introduced the graph, the graph is a major feature of the dashboard so it is centrally placed.

1.2.1 Iteration Three

This is the final dashboard wireframe which will be used as reference for the development of the dashboard. It shows general structure, ration and placement of items, with expanded menus to show as much design detail as possible (such as expanded search box).

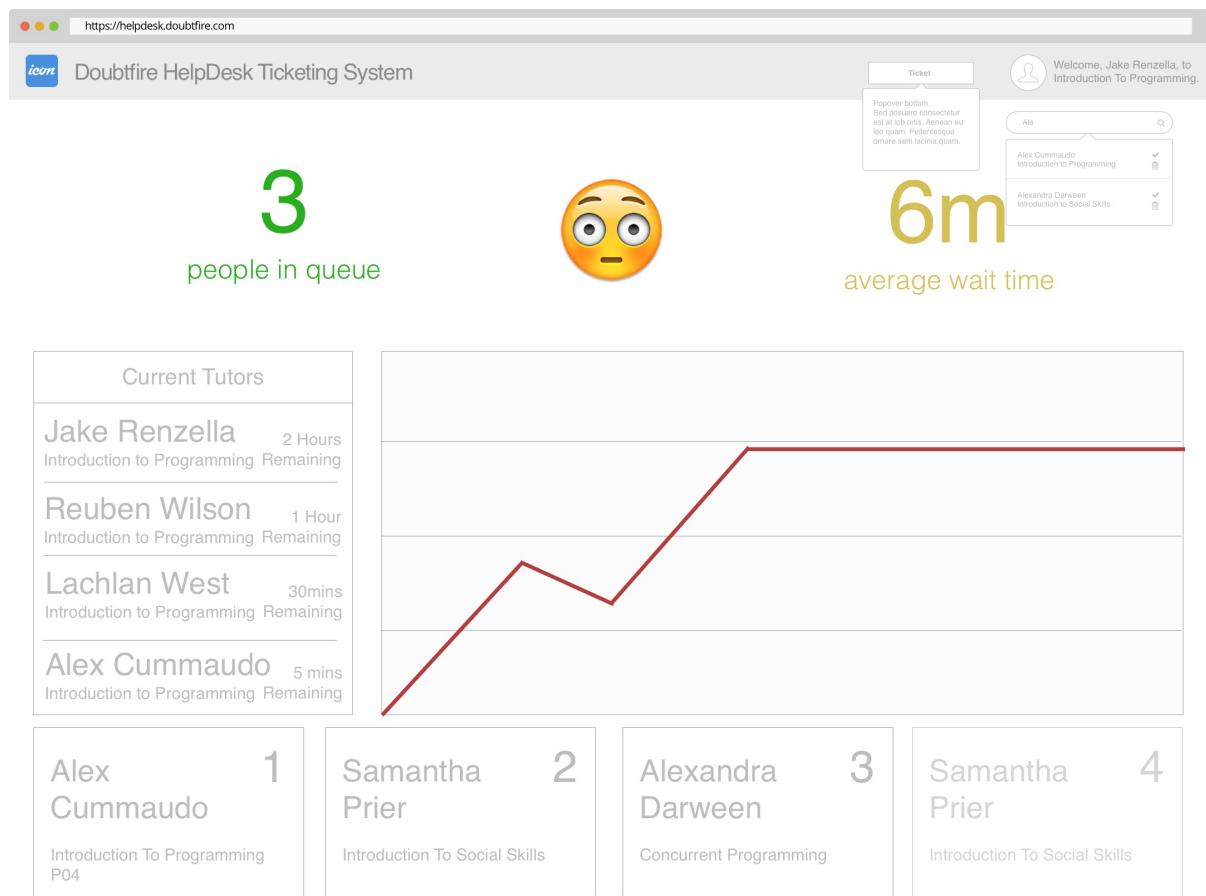
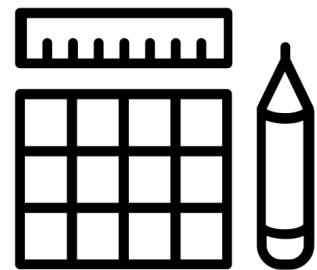


Figure 2: Iteration Three of the Dashboard

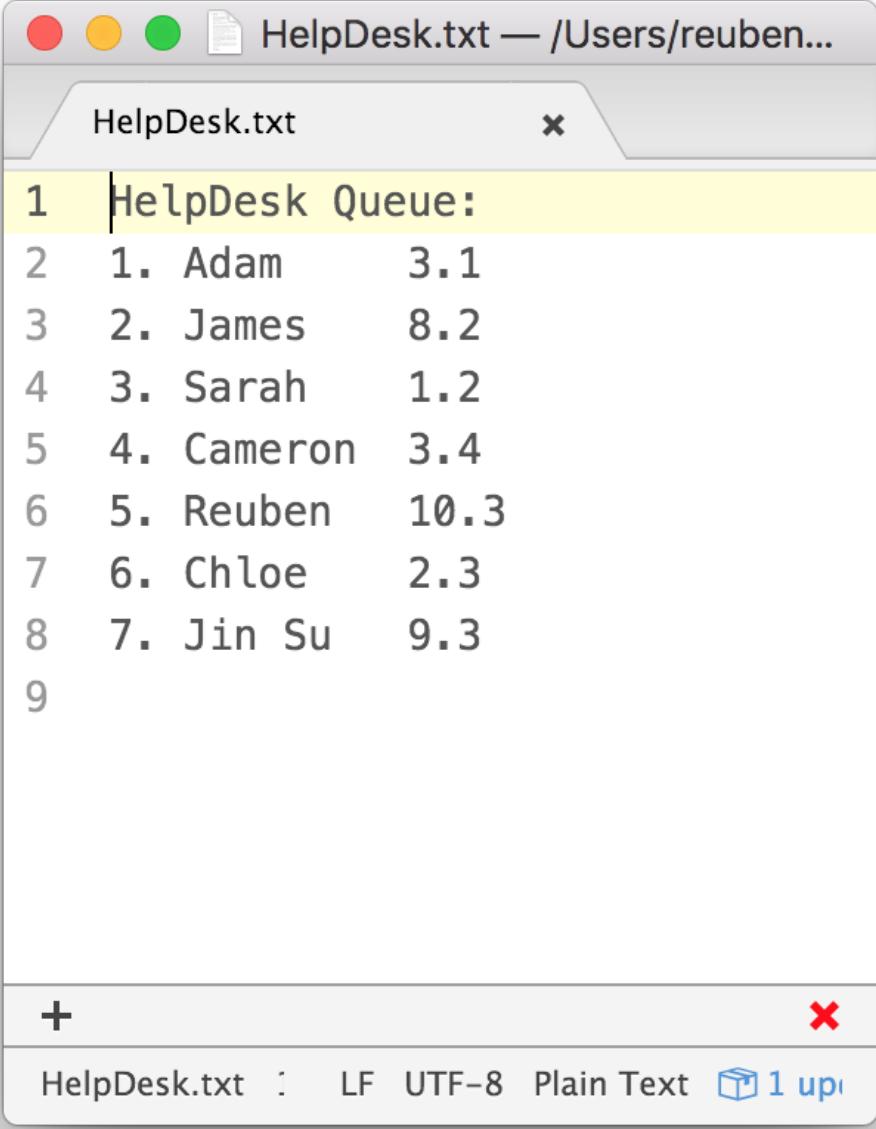


Chapter 8

Usability Testing Report

Contents

1 Purpose	2
2 Scope	4
2.1 Font Sizes	4
2.2 Basic Scenarios	4
2.3 Interpretation of dashboard	4
2.4 Mobile site navigation	4
2.5 User assumption	5
3 Test Strategy	5
3.1 Surveys	5
3.2 Live Testing with Participants	5
3.3 Post-Test Evaluation	5
4 Environment	6
5 Discussion of Results	6
5.1 Participant Demographics	6
5.2 Expectations of the Help Desk	6
5.3 Things Participant Liked about the Prototype	7
5.4 Things Participant Thought Could be Better about the Prototype	7
A General Instructions	8
B Pre-Evaluation Survey	11
C Raw Results	32
D User Evaluation Consent Forms	58



The screenshot shows a Mac OS X TextEdit window titled "HelpDesk.txt". The file path is "/Users/reuben...". The content of the file is a list of tickets in a queue:

Rank	Ticket Holder	Priority
1	Adam	3.1
2	James	8.2
3	Sarah	1.2
4	Cameron	3.4
5	Reuben	10.3
6	Chloe	2.3
7	Jin Su	9.3
9		

At the bottom of the window, there are buttons for "+" and "x", and status information: "HelpDesk.txt : LF UTF-8 Plain Text" and "1 up".

Figure 1: Previous Helpdesk System

2 Scope

What was tested:

2.1 Font Sizes

At various positions in the room, how well can subjects read various font sizes for text which is displayed on a projector. Refer to Figure 2.

Font A - O Romeo, Romeo! Wherefore art thou Romeo? - 10pt

Font B - Now is the winter of our discontent. - 12pt

Font C - A horse! a horse! my kingdom for a horse! - 14pt

Font D - Now is the winter of our discontent. - 16pt

Font E - To be, or not to be: that is the question. - 18pt

Font F - Is this a dagger which I see before me? - 20pt

Figure 2: Font size test

2.2 Basic Scenarios

Ask test subjects to perform various use scenarios to see if and how long it takes for someone to complete a particular task. Refer to the task descriptions in the appendix.

2.3 Interpretation of dashboard

How well were subjects able to gather information from the dashboard.

2.4 Mobile site navigation

How well were subjects able to navigate the mobile version of the system.

We are not testing:

2.5 User assumption

We assumed all users are familiar with basic PC input like mouse and keyboard, this is due to the nature of the end users of the system.

We assumed all users are familiar with basic web page navigation, menu styles and standards etc.

3 Test Strategy

The strategy chosen for usability testing will be in two parts. For both parts, the usability evaluator will be with the participant.

3.1 Surveys

Firstly the participants will fill out a survey, what they think about how the current helpdesk works and their ideal expectations. In addition, some demographic details will also be collected.

3.2 Live Testing with Participants

Secondly the participants will use a prototype of the helpdesk ticketing system and attempt to perform a number of defined tasks. The evaluator may help the participant if they get stuck on a task. During this time, the evaluator will make notes about any difficulties had, or comments made, by the participant. These details, and the surveys results, become the combined results of the evaluation.

3.3 Post-Test Evaluation

After the participant has completed the tasks, they will be asked to evaluate the prototype using the standard System Usability Scale questions. This will provide useful data on what could be improved with the prototype.

4 Environment

The test was held in ATC621, a similar environment to the actual helpdesk where the end system will be deployed. The reason it was held in this room is that we were not able to disturb the helpdesk. The test made use of:

- **A projector:** An Epson projector, same as which will be used in the HelpDesk.
- **PC:** Macbook Air with trackpad used to display the prototype.
- **Web Browser:** Google Chrome, the same as which will be used in the Helpdesk.
- **Mobile chrome view:** Used to simulate phone browser built into Google Chrome.
- **Keyboard/Mouse:** Same as which will be used in Helpdesk.

5 Discussion of Results

5.1 Participant Demographics

In terms of demographics, there was a limitation in that most participants were male and young. Although this does generally fit the demographics of people enrolled in programming units. The tutors that participated were all very experienced (4 or more semesters of tutoring).

5.2 Expectations of the Help Desk

Students were unanimous in being willing to wait 6 minutes or more for getting help in the help desk, which is more than what they report currently waiting. Student were generally not from using the helpdesk if the helpdesk was busy. The most helpful information that students wanted to know in advance before going to the help desk were: how many tutors are on duty, the units taught by the tutors and a simple description of how busy it is.

Tutors reported being able to support 4-5 students without being overburdened and they generally concurred with students in term of appropriate student wait time. The most helpful information tutors would like before helping a student were: the student's name and the unit and task they need help with.

The information desired by both students and tutors was generally in line with the interface of the prototype that the participants then went on to use.

5.3 Things Participant Liked about the Prototype

Participants were found to like the following attributes:

- Doing the major tasks like creating tickets, clocking on/off and viewing the queue did not present much difficulty to the participants.
- For students, being able to see which tutors are clocked on and what units they each was very useful to them.
- Tutors liked the clocking on/off feature
- Participants liked the mobile support of the dashboard.

In terms of the System Usability Scale evaluation results, the participants gave overwhelmingly positive feedback, with some exceptions. See the appendix for the full listing of results from the evaluations.

5.4 Things Participant Thought Could be Better about the Prototype

Participants were found not to like the following attributes

- Complex graph - some participants thought the graph showed too much information, the descriptive text of the axis was hard to read and/or it was not presented in a way that could convey the activity level of the help desk.
- Clock In Time - some participants found the decimal measure of hours (i.e. 0.75 meaning 45 minutes) not as intuitive as how fractions of hours are normally displayed.

A General Instructions

See attached document

General Instructions

Remember, this study is totally voluntary.

Although we don't know of any reason for this to happen, if you become uncomfortable or find this objectionable in any way, feel free to quit at any time.

We're testing the product, not you.

You're helping us by trying out this product. We're looking for places where the product may be difficult to use. If you have trouble with some of the tasks, it's the product's fault, not yours. Don't feel bad; that's exactly what we're looking for. If we can locate the trouble spots, then we can go back and improve the product.

Think Aloud

We have found that we get a great deal of information from these informal observations if we ask people to think aloud as they work through the exercises. It may be a bit awkward at first, but it's really very easy once you get used to it. All you have to do is speak your thoughts as you work.

You can perhaps think of this as "talking to yourself" about what's going on – what you are thinking about, what you are looking for on the screen, why you're looking for it, what you expect to see and to happen next, and so on. If you forget to think aloud, we'll remind you to keep talking.

Completing the Tasks

You will be asked to complete a number of tasks. Read the task out aloud. When you think you understand the task please say 'Ready'. We will then ask you to start the task. When you have finished please let us know by saying 'Finished'. We may ask you to start the next task before you have finished your current task.

When this happens, it does not mean that you have failed to complete the task. Sometimes we have learned all we need about how the product works for that task. We would rather go on to the next task than to waste your time.

Questions?

If you feel you need help with something, please raise your hand and we will attempt to answer your question.

B Pre-Evaluation Survey

See attached document

Pre-Evaluation Survey

Please answer all the following questions so we can find out a few things about you.

This helps us interpret our results.

*Required

Demographics

1. Select the following age group that includes your own age. *

Mark only one oval.

- 18 to 24
- 25 to 34
- 35 to 49
- 50 or over

2. Select your gender. *

Mark only one oval.

- Male
- Female
- Rather not say

Helpdesk Attendance

3. How many hours, on average, do you attend the helpdesk? *

Mark only one oval.

- Less than an hour
- An hour to two hours
- Two hours to four hours
- More than four hours

4. How many days a week, on average, do you attend the helpdesk? *

Mark only one oval.

- Only once a week
- Once or twice a week
- Three to four times a week
- Every day in a week

Staff Questions

Please complete this section only if you are a tutor at the helpdesk.

6. How many semesters have you taught at the helpdesk? *

Mark only one oval.

- One semester
- Two to three semesters
- Three to four semesters
- More than four semesters

7. Please check all the subjects you teach as an employed Swinburne tutor, if any. *

Tick all that apply.

- Introduction to Programming
- Object Oriented Programming
- Creating Web Applications
- Other:

8. How many students to concurrently support without being overburdened? *

Mark only one oval.

- More than two students
- More than three students
- More than four students
- More than five students
- More than six students

9. How long do you think is acceptable for students to wait? *

Mark only one oval.

- More than two minutes
- More than three minutes
- More than four minutes
- More than five minutes
- More than six minutes

10. **What three pieces of information would be most helpful for you before you see a student for help? ***

Tick all that apply.

- The student's name
- The student's photo
- The task they need help with
- The unit they need help with
- A description outlining their problem
- Other students who also have a similar issue
- Other:

Skip to question 15.

Student Questions

Please complete this section only if you are a student.

11. What is the maximum time you would be happy waiting for assistance? *

Mark only one oval.

- More than two minutes
- More than three minutes
- More than four minutes
- More than five minutes
- More than six minutes

12. How long, on average, do you have to wait for assistance now? *

Mark only one oval.

- More than two minutes
- More than three minutes
- More than four minutes
- More than five minutes
- More than six minutes

13. If you could see how busy the helpdesk is before going there, would it affect the likelihood of going there. *

Mark only one oval.

- Yes, definitely
- Maybe
- No I would still try and get help

14. Before going to the helpdesk, what three pieces of information would be helpful before going? *

Tick all that apply.

- How many staff are working there
- The units taught by the tutors
- How many students are currently waiting
- The current average waiting times for students
- The trend of average waiting times for students over the last three hours
- The trend of how many students have been waiting over the last three hours
- A one-word description/icon describing how busy the helpdesk is
- Other:

Skip to question 15.

Font Testing

For these questions please stand at the back of the room and look at the projector.

Each font is listed A to F from top to bottom.

15. Rate each font on the screen based on how well you could read the sentence. *

Mark only one oval per row.

I can't read it all I can just read it I can read it I can read it well

Font A - 10pt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Font B - 12pt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Font C - 14pt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Font D - 16pt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Font E - 18pt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Font F - 20pt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

16. Do you have any issues with your eyesight? *

Mark only one oval.

- No
- Yes - short sightedness
- Yes - long sightedness
- Yes - colour blindness
- Yes - other

Post-Evaluation Survey

At this point we ask that you review the dashboard prototype displayed on the projector.

Please answer the following questions based on your evaluation onscreen.

17. List two things you liked about the dashboard. *

.....
.....
.....
.....
.....

18. List two things you would change about the dashboard. *

.....
.....
.....
.....
.....

19. Could you read all information on the screen? *

Mark only one oval.

- Yes
 No

20. If you answered no to the previous question, please list what you could not read.

.....
.....
.....
.....
.....

21. Could you interpret what the graph was displaying? *

Mark only one oval.

- Yes
 No

22. Would you prefer to sign in to access this information? *

Mark only one oval.

- Yes
- No

23. When a ticket is resolved, would you like to be notified by a visual or audio cue?

*
Mark only one oval.

- Visual cue (e.g., screen flash)
- Audio cue (e.g., ping noise)
- Both
- None

24. Do you think such a dashboard would entice you to go to the helpdesk more often (if you are a student) or help you with assisting students (if you are a staff member)? *

Mark only one oval.

- Yes
- No

25. Please note any additional comments you would like to make about the dashboard you have been shown.

.....
.....
.....
.....
.....

Extended Evaluation

26. Would you be interested in helping us with an extended evaluation of the Helpdesk ticketing system? *

Mark only one oval.

Yes

No *Please submit the form to the evaluator.*

User Task Descriptions

Please refer to the general instructions sheet provided before continuing.

Once you have read these instructions your facilitator will guide you through to the evaluation process.

Please only continue to the next section after you have finished each of the tasks asked of you.

Task 1

Task Description

In this task we would like you to put yourself in the shoes of a student who wants to attend the helpdesk. You are enrolled in two subjects:

1. COS30243 - Game Programming
2. COS20007 - Object Oriented Programming

You are having trouble with Object Oriented Programming, more specifically with the following compiler error you are getting on Task A16:

clang++: could not load type from assembly

Please attempt to create a ticket for COS20007 under Task A6, providing the above text as the description for the ticket.

Information Needed

In order to sign in to Doubtfire as the student, use the following credentials:

- Username is: astudent
- Password is: password

27. **Were you able to successfully create your ticket? ***

Mark only one oval.

Yes

No

I don't know

After finishing this task...

After creating this ticket, please attempt to navigate to the Helpdesk Dashboard and answer the questions below.

28. What is your ticket number in the queue? *

.....

29. Is there a tutor currently working for the unit you put on your ticket? *

Mark only one oval.

Yes

No

I don't know

30. Based on the graph alone, when was the best time to seek assistance at the helpdesk? *

Mark only one oval.

Zero to one hours ago

One to two hours ago

Two to three hours ago

31. Based on the graph alone, when was the worst time to seek assistance at the helpdesk? *

Mark only one oval.

Zero to one hours ago

One to two hours ago

Two to three hours ago

Task 2

Task Description

Continuing on from the scenario described in Task 1, as you are waiting at the helpdesk, you eventually resolve your issue before a tutor has come around to assist you.

Please attempt to close the ticket you created in Task 1.

32. Were you able to close your ticket successfully? *

Mark only one oval.

Yes

No

I don't know

After finishing this task...

Once you complete this task you will need to sign out of this Doubtfire account.

Please do so by clicking the username in the top-right corner of the screen, and then click "Sign Out".

Task 3

Task Description

In this task we would like you to put yourself in the shoes of a tutor who is about to begin working at the helpdesk. The tutor teaches the following unit:

- COS20007 - Object Oriented Programming

You are scheduled to work for 2h and 45mins.

Please attempt to clock on to the helpdesk for this scheduled work time.

Information Needed

In order to sign in to Doubtfire as the student, use the following credentials:

- Username is: atutor
- Password is: password

33. Were you able to clock on at the Helpdesk for the scheduled 2 hours and 45 minutes? *

Mark only one oval.

Yes

No

I don't know

After finishing this task...

After creating this ticket, please attempt to navigate to the Helpdesk Dashboard and answer the questions below.

34. Based on the dashboard data alone, are you able to determine the exact time when you will automatically be clocked off? *

Mark only one oval.

Yes

No

I don't know

Task 4

Before you begin...

Please allow the facilitator to switch the browser into a simulated smartphone view.

Task Description

Continuing on from the description described in Task 3, we would like you to imagine you are walking around the helpdesk with your smartphone.

Please attempt to resolve any unresolved tickets for the unit COS20007 - Object Oriented Programming.

35. **Were you able to find an unresolved ticket for the unit COS20007 - Object Oriented Programming? ***

Mark only one oval.

Yes

No

I don't know

36. **If you answered yes for the previous question, were you able to resolve the ticket?**

Mark only one oval.

Yes

No

I don't know

Task 5

Task Description

Continuing on from the scenario described in Tasks 3 and 4, you receive a phone call in the middle of your shift and you need to leave the helpdesk immediately to attend an emergency tutor meeting.

Please attempt to clock off.

37. Were you able to clock off successfully? *

Mark only one oval.

Yes

No

I don't know

Extended Post-Evaluation Survey

Please only complete this section only if you have completed the user task descriptions.

38. On a scale of 1 to 4, how familiar you are with using Doubtfire? *

Mark only one oval.

1 2 3 4

Not very familiar Very familiar

39. System Usability Scale *

Please tick the option that best represents your reaction to the ticketing system. Don't think too hard about each question. We are interested in your first reaction. The System Usability Scale is © Digital Equipment Corporation, 1986.

Mark only one oval per row.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I think that I would like to use this system frequently.	<input type="radio"/>				
I found the system unnecessarily complex.	<input type="radio"/>				
I thought the system was easy to use.	<input type="radio"/>				
I think that I would need the support of a technical person to be able to use this system.	<input type="radio"/>				
I found the various functions in this system were well integrated.	<input type="radio"/>				
I thought there was too much inconsistency in this system.	<input type="radio"/>				
I would imagine that most people would learn to use this system very quickly.	<input type="radio"/>				
I found the system very cumbersome to use.	<input type="radio"/>				
I felt very confident using the system.	<input type="radio"/>				
I needed to learn a lot of things before I could get going with this system.	<input type="radio"/>				

40. Please list two things that you most liked about the prototype. *

.....
.....
.....
.....
.....

41. Please list two things that you least liked about the prototype.*

.....
.....
.....
.....
.....

42. Qualitative Descriptions

Please select words that you think apply to the prototype.

Tick all that apply.

- Ugly
- Effortless
- Unnatural
- Simple
- Efficient
- Consistent
- Hidden
- Frustrating
- Easy
- Visible
- Irregular
- Slow
- Complicated
- Intuitive
- Difficult
- Attractive

43. Task Difficulty *

Mark only one oval per row.

	Very Easy	Easy	Hard	Very Hard
Task 1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Task 2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Task 3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Task 4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Task 5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

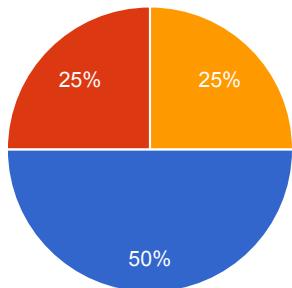
C Raw Results

See attached document

Results

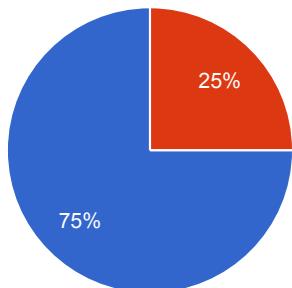
Demographics

Select the following age group that includes your own age.



18 to 24	2	50%
25 to 34	1	25%
35 to 49	1	25%
50 or over	0	0%

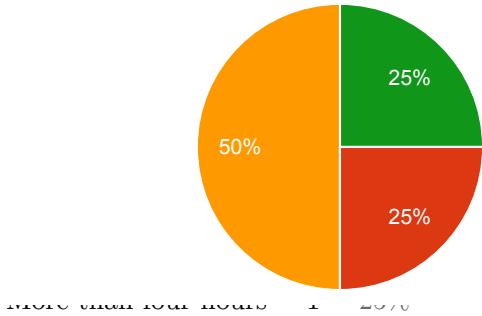
Select your gender.



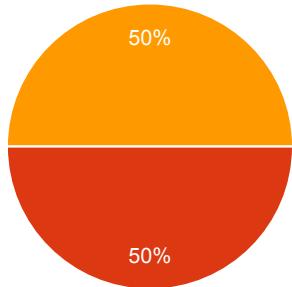
Male	3	75%
Female	1	25%
Rather not say	0	0%

Helpdesk Attendance

How many hours, on average, do you attend the helpdesk?

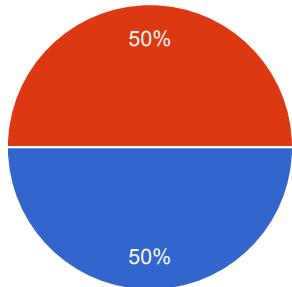


How many days a week, on average, do you attend the helpdesk?



Only once a week	0	0%
Once or twice a week	2	50%
Three to four times a week	2	50%
Every day in a week	0	0%

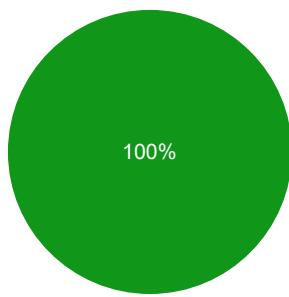
Do you attend the helpdesk as a staff member or as a student?



A staff member	2	50%
A student	2	50%

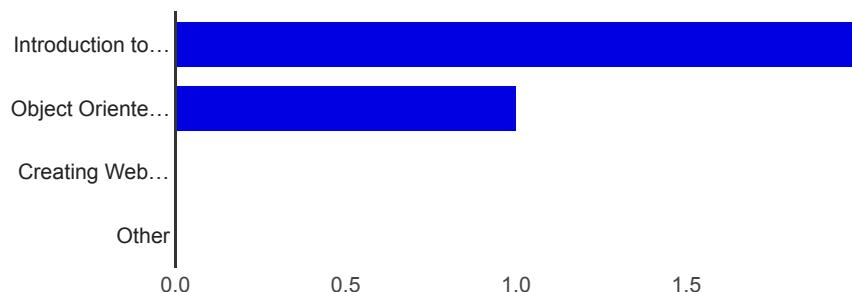
Staff Questions

How many semesters have you taught at the helpdesk?



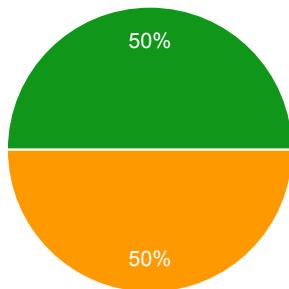
One semester	0	0%
Two to three semesters	0	0%
Three to four semesters	0	0%
More than four semesters	2	100%

Please check all the subjects you teach as an employed Swinburne tutor, if any.



Introduction to Programming	2	100%
Object Oriented Programming	1	50%
Creating Web Applications	0	0%
Other	0	0%

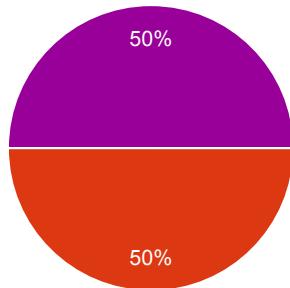
How many students to concurrently support without being overburdened?



More than two students	0	0%
More than three students	0	0%
More than four students	1	50%

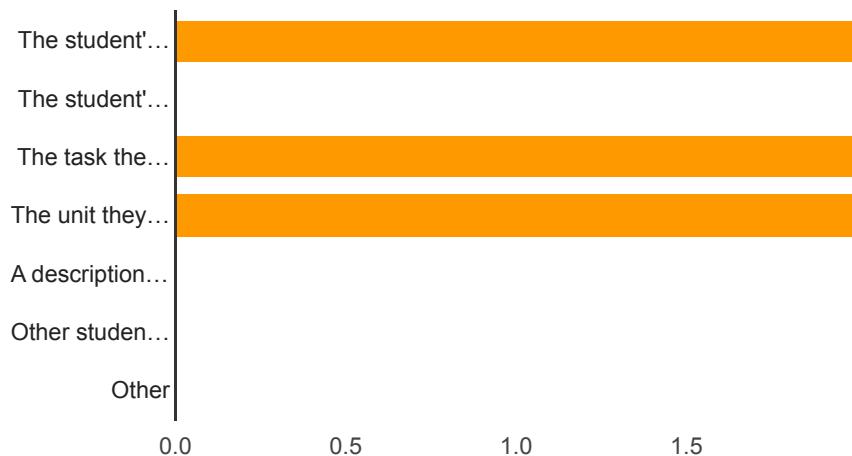
More than five students	1	50%
More than six students	0	0%

How long do you think is acceptable for students to wait?



More than two minutes	0	0%
More than three minutes	1	50%
More than four minutes	0	0%
More than five minutes	0	0%
More than six minutes	1	50%

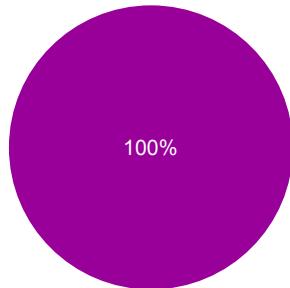
What three pieces of information would be most helpful for you before you see a student for help?



The student's name	2	100%
The student's photo	0	0%
The task they need help with	2	100%
The unit they need help with	2	100%
A description outlining their problem	0	0%
Other students who also have a similar issue	0	0%
Other	0	0%

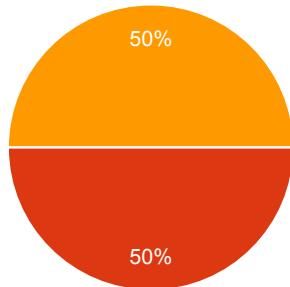
Student Questions

What is the maximum time you would be happy waiting for assistance?



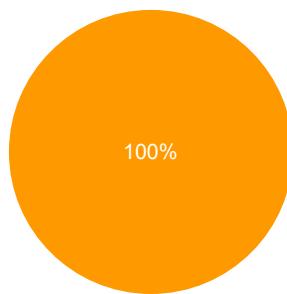
More than two minutes	0	0%
More than three minutes	0	0%
More than four minutes	0	0%
More than five minutes	0	0%
More than six minutes	2	100%

How long, on average, do you have to wait for assistance now?



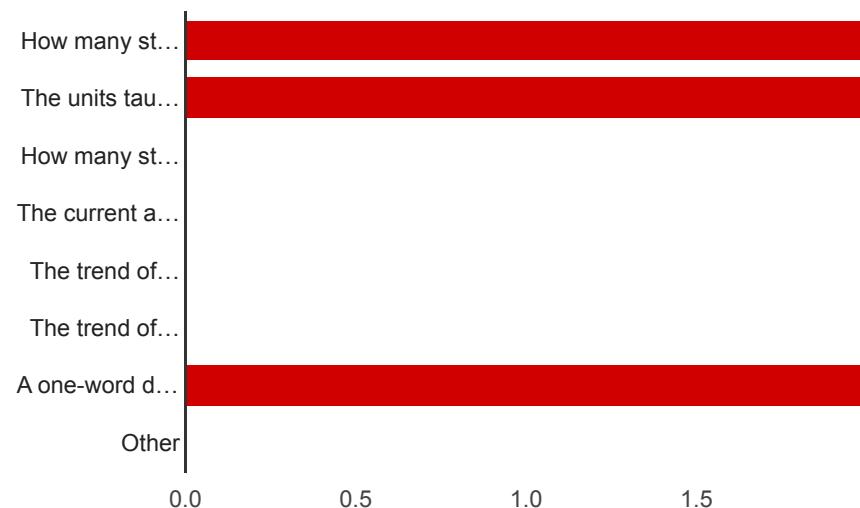
More than two minutes	0	0%
More than three minutes	1	50%
More than four minutes	1	50%
More than five minutes	0	0%
More than six minutes	0	0%

If you could see how busy the helpdesk is before going there, would it affect the likelihood of going there.



Yes, definitely	0	0%
Maybe	0	0%
No I would still try and get help	2	100%

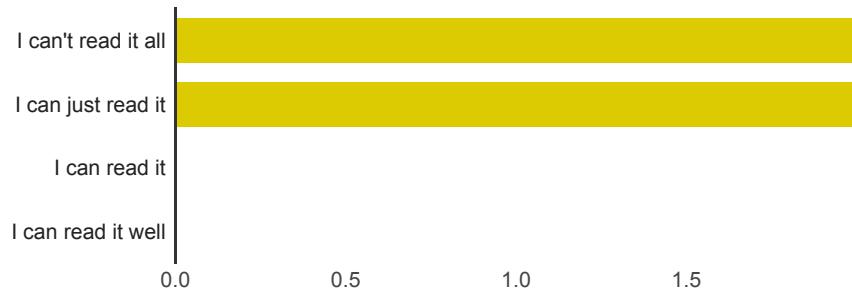
Before going to the helpdesk, what three pieces of information would be helpful before going?



How many staff are working there	2	100%
The units taught by the tutors	2	100%
How many students are currently waiting	0	0%
The current average waiting times for students	0	0%
The trend of average waiting times for students over the last three hours	0	0%
The trend of how many students have been waiting over the last three hours	0	0%
A one-word description/icon describing how busy the helpdesk is	2	100%
Other	0	0%

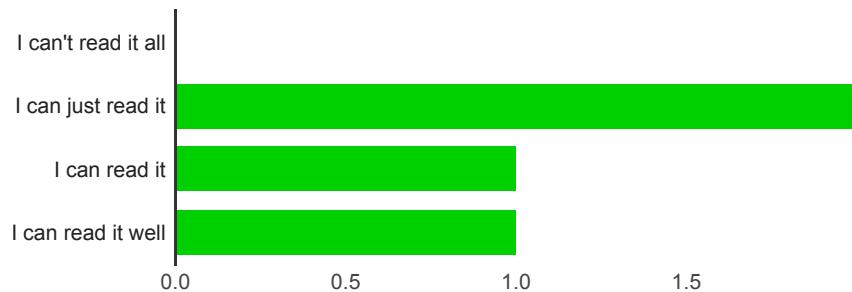
Font Testing

Font A - 10pt [Rate each font on the screen based on how well you could read the sentence.]



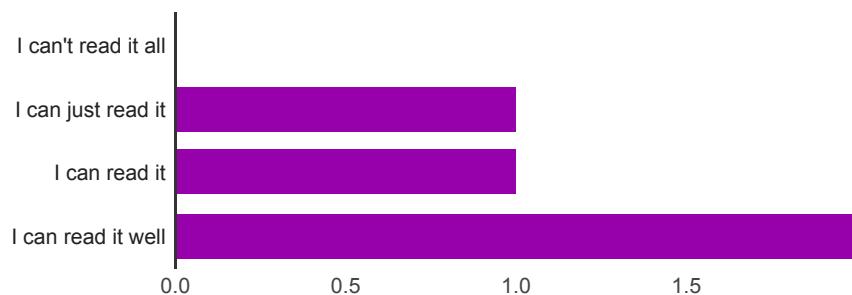
I can't read it all	2	50%
I can just read it	2	50%
I can read it	0	0%
I can read it well	0	0%

Font B - 12pt [Rate each font on the screen based on how well you could read the sentence.]



I can't read it all	0	0%
I can just read it	2	50%
I can read it	1	25%
I can read it well	1	25%

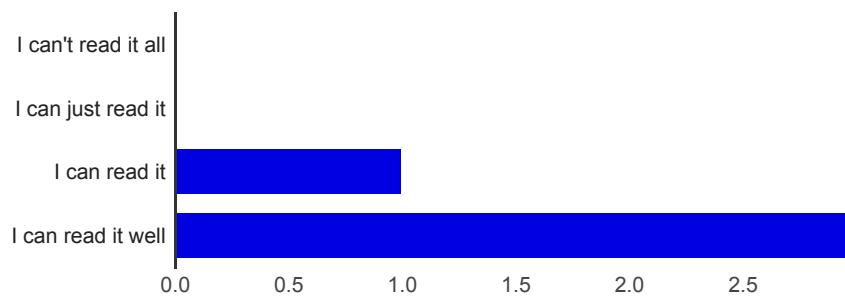
Font C - 14pt [Rate each font on the screen based on how well you could read the sentence.]



I can't read it all	0	0%
I can just read it	1	25%
I can read it	1	25%
I can read it well	2	50%

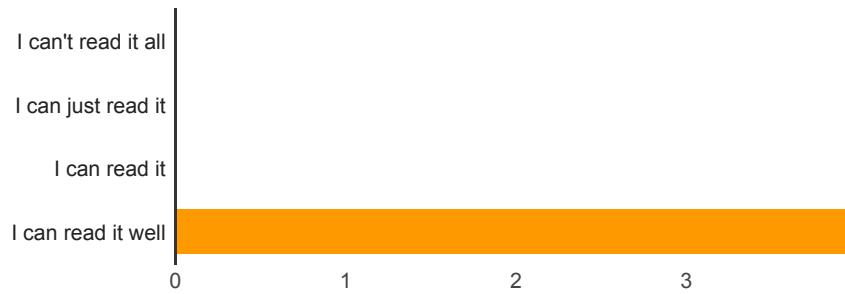
I can just read it	1	25%
I can read it	1	25%
I can read it well	2	50%

Font D - 16pt [Rate each font on the screen based on how well you could read the sentence.]



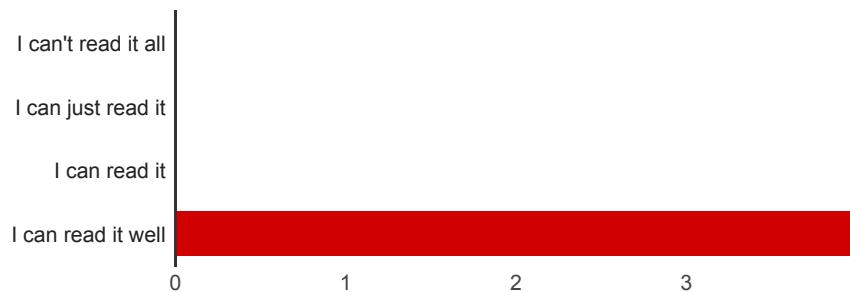
I can't read it all	0	0%
I can just read it	0	0%
I can read it	1	25%
I can read it well	3	75%

Font E - 18pt [Rate each font on the screen based on how well you could read the sentence.]

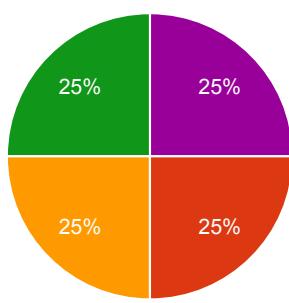


I can't read it all	0	0%
I can just read it	0	0%
I can read it	0	0%
I can read it well	4	100%

Font F - 20pt [Rate each font on the screen based on how well you could read the sentence.]



Do you have any issues with your eyesight?



No	0	0%
Yes - short sightedness	1	25%
Yes - long sightedness	1	25%
Yes - colour blindness	1	25%
Yes - other	1	25%

Post-Evaluation Survey

List two things you liked about the dashboard.

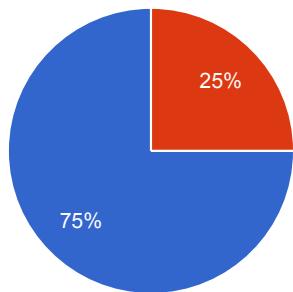
- lots of info, easy to see tickets for helping
- detailed, like to know who is on
- big numbers for people in queue, who's working but what's working
- Graph, tutors are on the lefthand side

List two things you would change about the dashboard.

- areas could be more well defined, average wait time graph might be unnecessary
- don't care about previous hour, hard to decipher

unit names instead of codes for tutors working, tutor picture, graph distracting
nothing

Could you read all information on the screen?

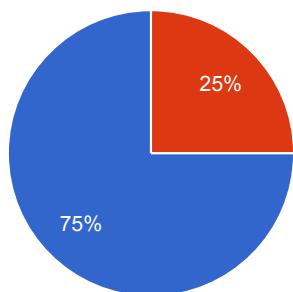


Yes	3	75%
No	1	25%

If you answered no to the previous question, please list what you could not read.

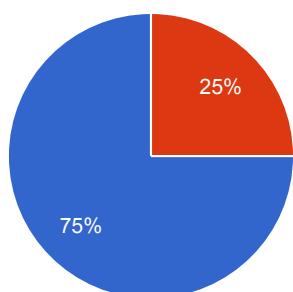
Axis on graph

Could you interpret what the graph was displaying?



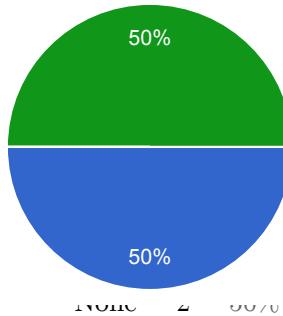
Yes	3	75%
No	1	25%

Would you prefer to sign in to access this information?

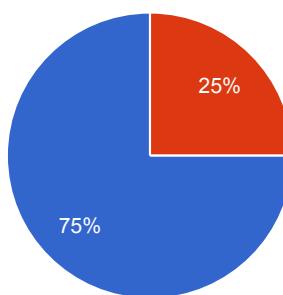


Yes	3	75%
No	1	25%

When a ticket is resolved, would you like to be notified by a visual or audio cue?



Do you think such a dashboard would entice you to go to the helpdesk more often (if you are a student) or help you with assisting students (if you are a staff member)?



Yes 3 75%

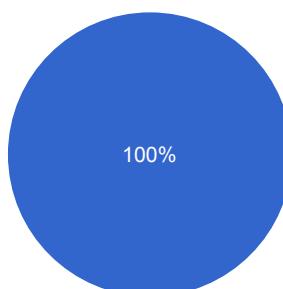
No 1 25%

Please note any additional comments you would like to make about the dashboard you have been shown.

3 hours at once should be last 30 minutes
hide the graph on a separate tab perhaps

Extended Evaluation

Would you be interested in helping us with an extended evaluation of the Helpdesk ticketing system?



Yes 4 100%

No 0 0%

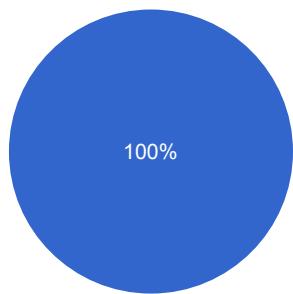
User Task Descriptions

Task 1

Task Description

Information Needed

Were you able to successfully create your ticket?



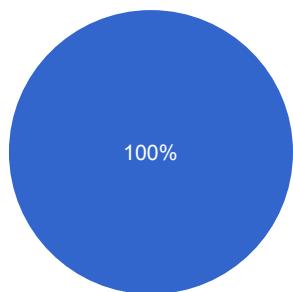
Yes	4	100%
No	0	0%
I don't know	0	0%

After finishing this task...

What is your ticket number in the queue?

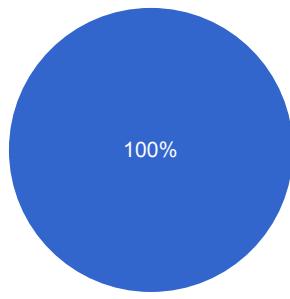
4

Is there a tutor currently working for the unit you put on your ticket?



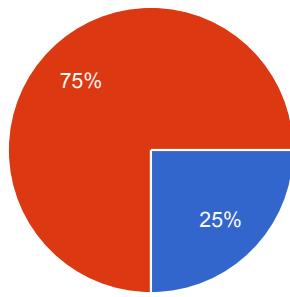
Yes	4	100%
No	0	0%
I don't know	0	0%

Based on the graph alone, when was the best time to seek assistance at the helpdesk?



Two to three hours ago 0 0%

Based on the graph alone, when was the worst time to seek assistance at the helpdesk?



Zero to one hours ago 1 25%

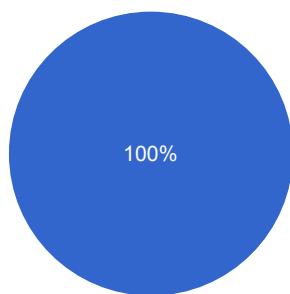
One to two hours ago 3 75%

Two to three hours ago 0 0%

Task 2

Task Description

Were you able to close your ticket successfully?



Yes 4 100%

No 0 0%

I don't know 0 0%

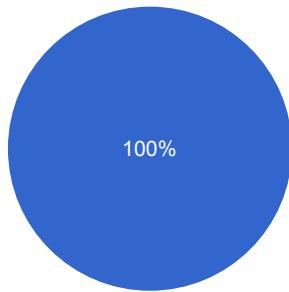
After finishing this task...

Task 3

Task Description

Information Needed

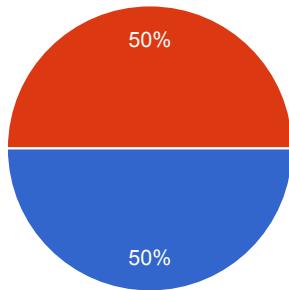
Were you able to clock on at the Helpdesk for the scheduled 2 hours and 45 minutes?



Yes	4	100%
No	0	0%
I don't know	0	0%

After finishing this task...

Based on the dashboard data alone, are you able to determine the exact time when you will automatically be clocked off?



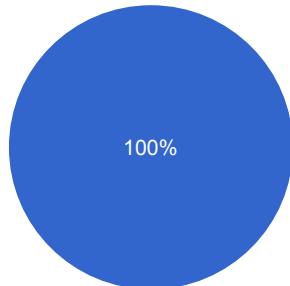
Yes	2	50%
No	2	50%
I don't know	0	0%

Task 4

Before you begin...

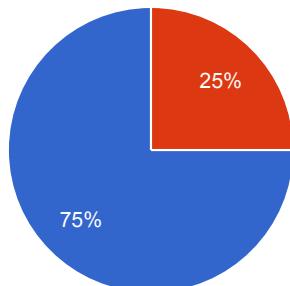
Task Description

Were you able to find an unresolved ticket for the unit COS20007 - Object Oriented Programming?



Yes	4	100%
No	0	0%
I don't know	0	0%

If you answered yes for the previous question, were you able to resolve the ticket?

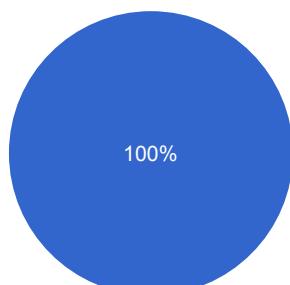


Yes	3	75%
No	1	25%
I don't know	0	0%

Task 5

Task Description

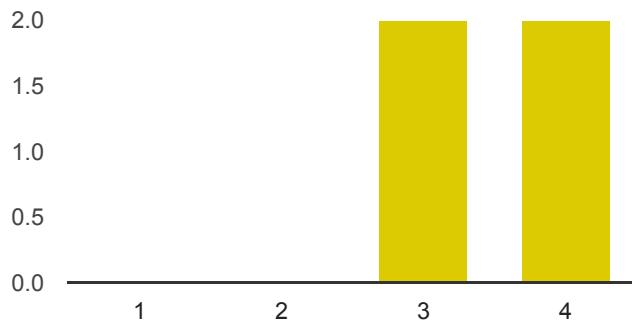
Were you able to clock off successfully?



Yes	4	100%
No	0	0%
I don't know	0	0%

Extended Post-Evaluation Survey

On a scale of 1 to 4, how familiar you are with using Doubtfire?



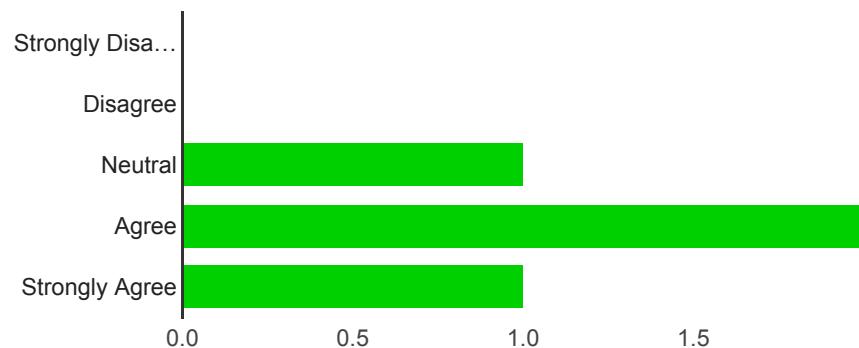
Not very familiar: 1 0 0%

 2 0 0%

 3 2 50%

Very familiar: 4 2 50%

I think that I would like to use this system frequently. [System Usability Scale]



Strongly Disagree 0 0%

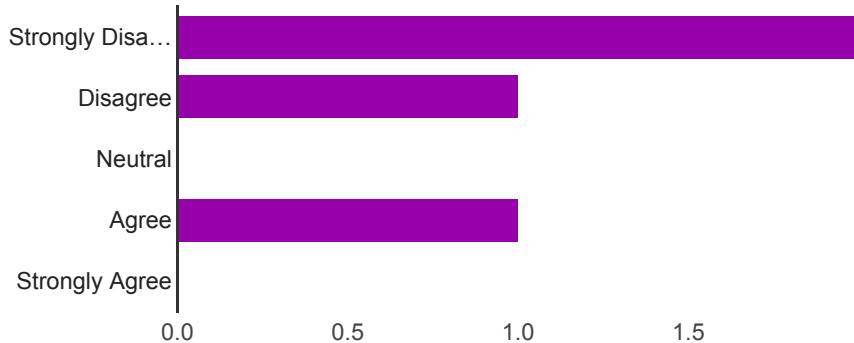
Disagree 0 0%

Neutral 1 25%

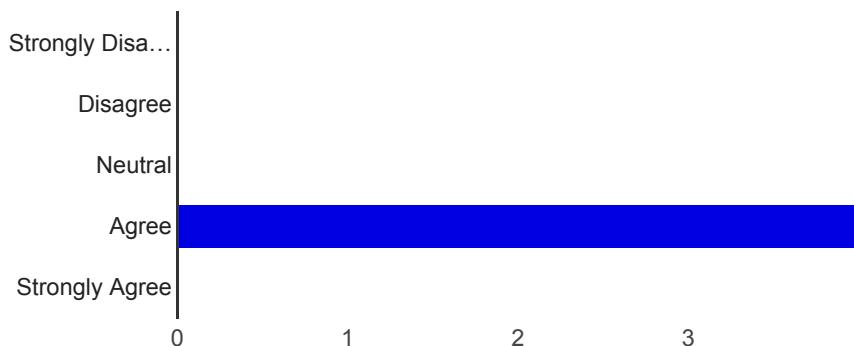
Agree 2 50%

Strongly Agree 1 25%

I found the system unnecessarily complex. [System Usability Scale]

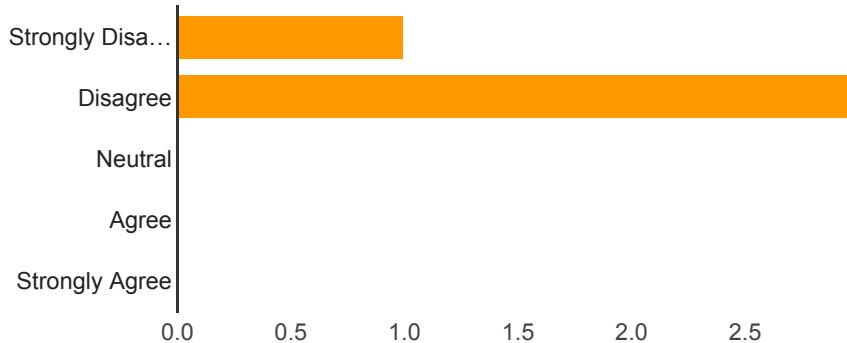


I thought the system was easy to use. [System Usability Scale]

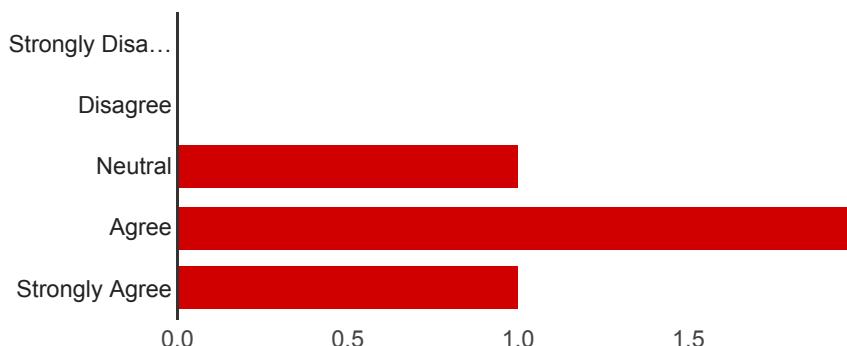


Strongly Disagree 0 0%
Disagree 0 0%
Neutral 0 0%
Agree 4 100%
Strongly Agree 0 0%

I think that I would need the support of a technical person to be able to use this system. [System Usability Scale]

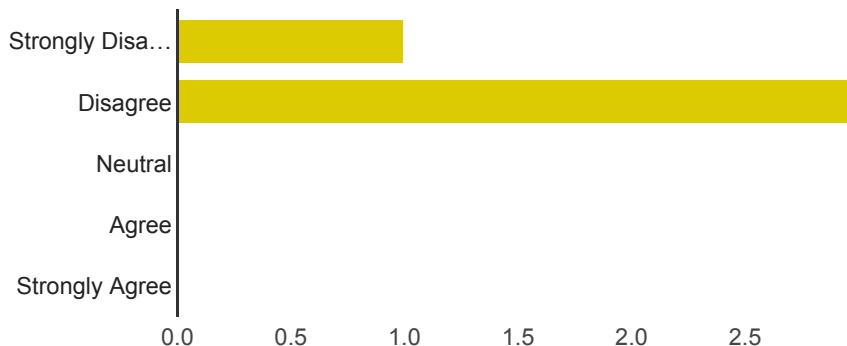


I found the various functions in this system were well integrated. [System Usability Scale]



Strongly Disagree	0	0%
Disagree	0	0%
Neutral	1	25%
Agree	2	50%
Strongly Agree	1	25%

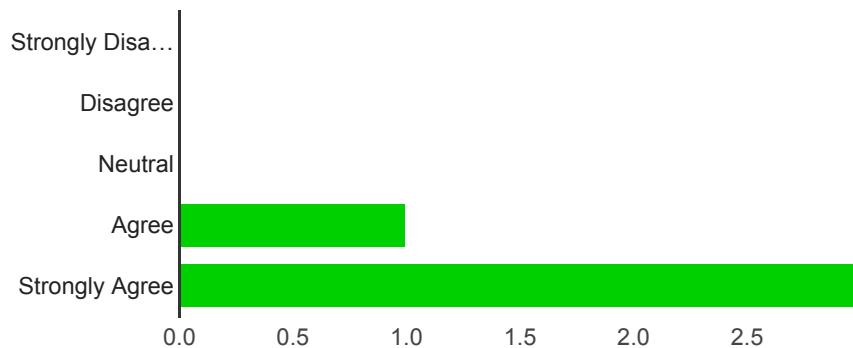
I thought there was too much inconsistency in this system. [System Usability Scale]



Strongly Disagree	1	25%
Disagree	3	75%
Neutral	0	0%

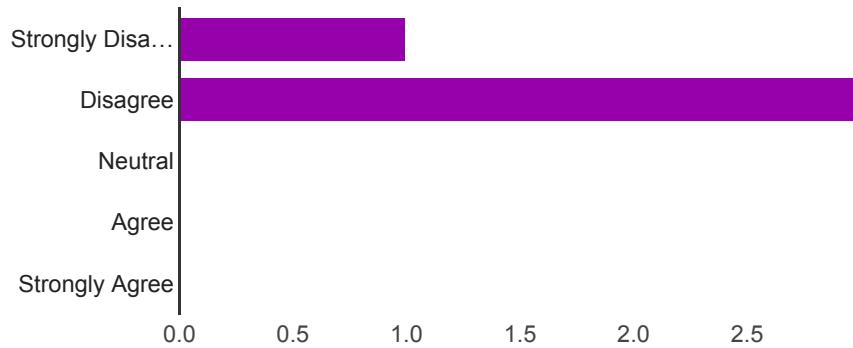
Agree	0	0%
Strongly Agree	0	0%

I would imagine that most people would learn to use this system very quickly.
[System Usability Scale]



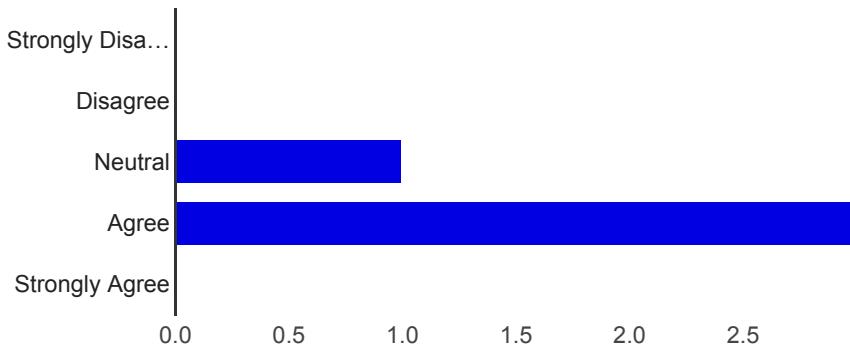
Strongly Disagree	0	0%
Disagree	0	0%
Neutral	0	0%
Agree	1	25%
Strongly Agree	3	75%

I found the system very cumbersome to use. [System Usability Scale]



Strongly Disagree	1	25%
Disagree	3	75%
Neutral	0	0%
Agree	0	0%
Strongly Agree	0	0%

I felt very confident using the system. [System Usability Scale]



Strongly Disagree 0 0%

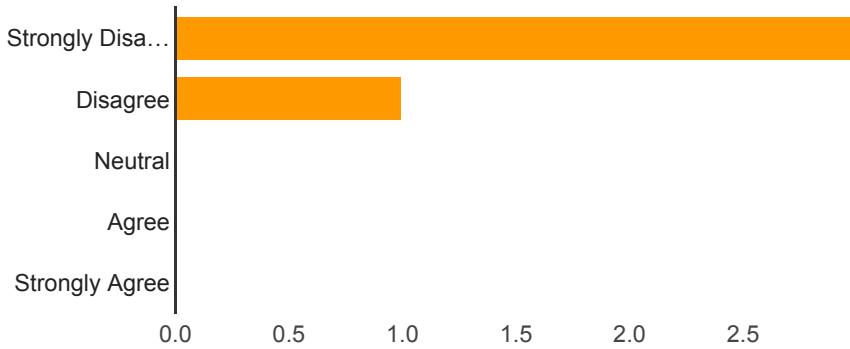
Disagree 0 0%

Neutral 1 25%

Agree 3 75%

Strongly Agree 0 0%

I needed to learn a lot of things before I could get going with this system. [System Usability Scale]



Strongly Disagree 3 75%

Disagree 1 25%

Neutral 0 0%

Agree 0 0%

Strongly Agree 0 0%

Please list two things that you most liked about the ticketing system.

Phone support, easy to see tickets

ability to access help without being passed by, can tell if the tutors are doing my subject

Clocking on/off is a good feature; seeing which tutors are working

interactivity, easy ability to view the useful information

Please list two things that you least liked about the ticketing system.

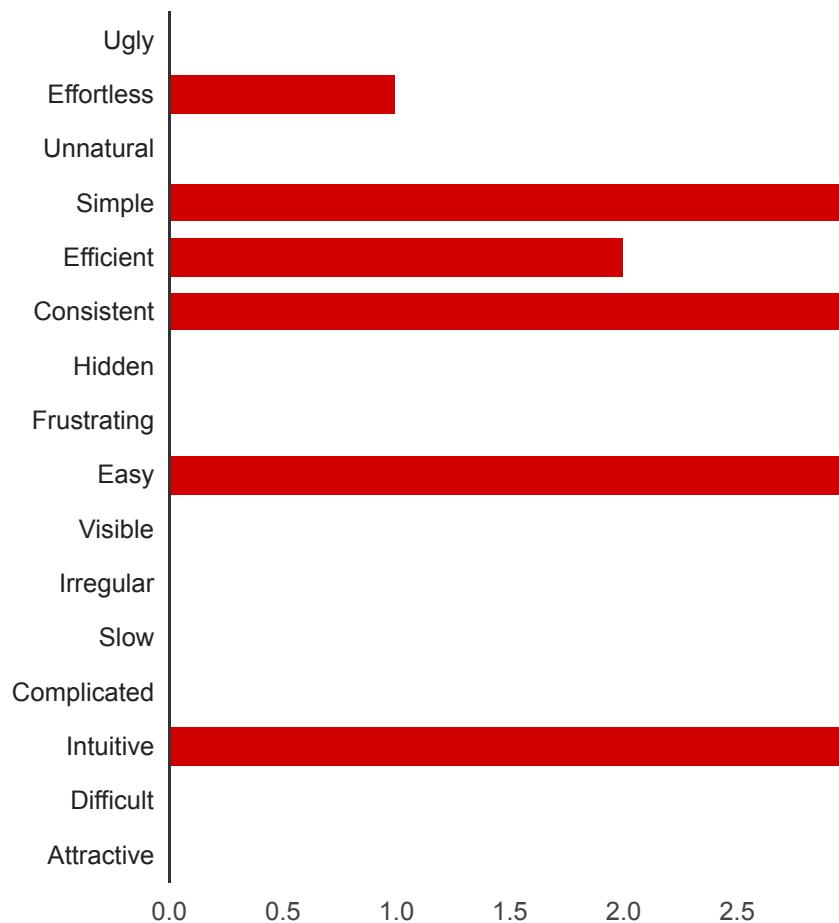
clock in time

graphs were hard to decipher, had to guess where the poignant icons were based on previous systems I have used.

Graph is too large and possibly unnecessary; not sure if students should be able to see other's tickets

n/a

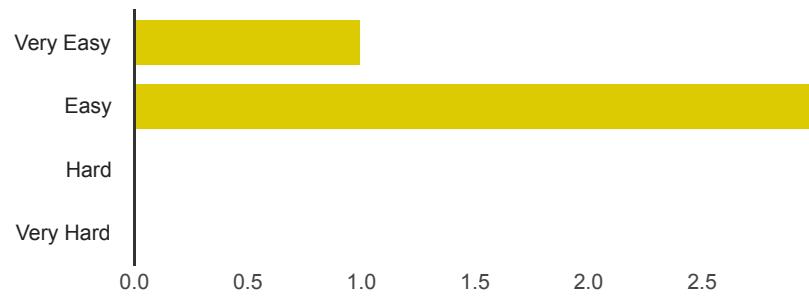
Qualitative Descriptions



Ugly	0	0%
Effortless	1	25%
Unnatural	0	0%
Simple	3	75%
Efficient	2	50%
Consistent	3	75%
Hidden	0	0%

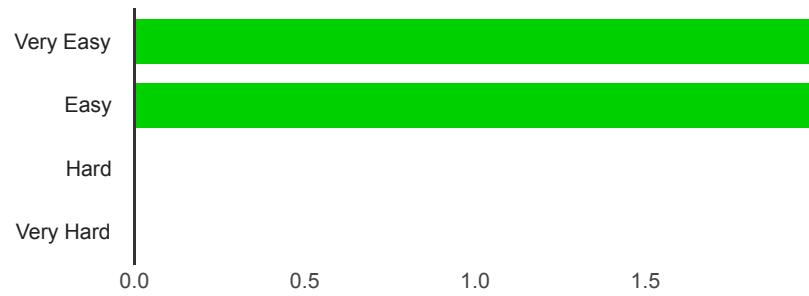
Frustrating	0	0%
Easy	3	75%
Visible	0	0%
Irregular	0	0%
Slow	0	0%
Complicated	0	0%
Intuitive	3	75%
Difficult	0	0%
Attractive	0	0%

Task 1 - Creating Ticket [Task Difficulty]



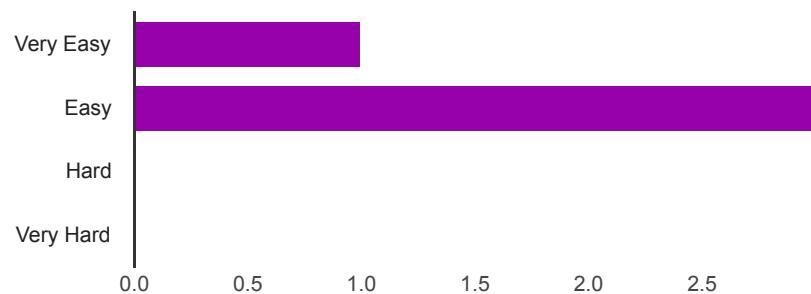
Very Easy	1	25%
Easy	3	75%
Hard	0	0%
Very Hard	0	0%

Task 2 - Closing Ticket [Task Difficulty]



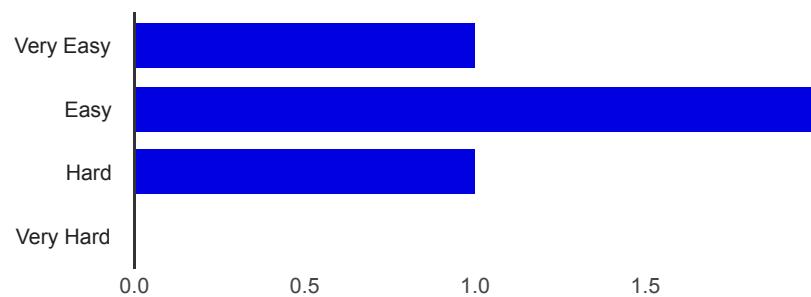
Very Easy	2	50%
Easy	2	50%
Hard	0	0%
Very Hard	0	0%

Task 3 - Clocking On [Task Difficulty]



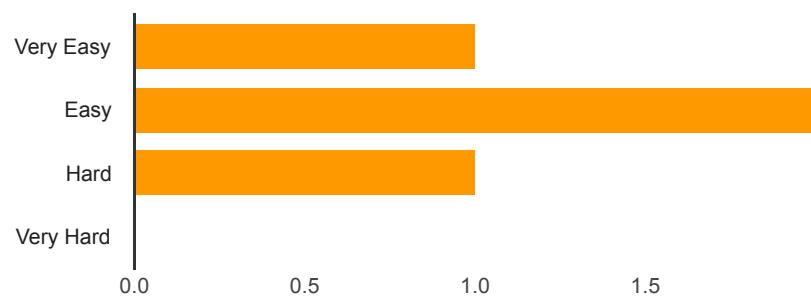
Difficulty Level	Count	Percentage
Very Easy	1	25%
Easy	3	75%
Hard	0	0%
Very Hard	0	0%

Task 4 - Resolving Ticket [Task Difficulty]



Difficulty Level	Count	Percentage
Very Easy	1	25%
Easy	2	50%
Hard	1	25%
Very Hard	0	0%

Task 5 - Clocking Off [Task Difficulty]



Difficulty Level	Count	Percentage
Very Easy	1	25%
Easy	3	75%
Hard	0	0%
Very Hard	0	0%

Easy	2	50%
Hard	1	25%
Very Hard	0	0%

D User Evaluation Consent Forms

See attached document



Chapter 9

Technical Manual

Contents

1 Doubtfire Git Workflow	2
1.1 Table of Contents	2
1.2 About the Doubtfire Branch Structure	2
1.3 Getting started with the Forking Workflow	5
1.3.1 1. Forking and Cloning the repository	5
1.3.2 2. Writing your new changes	7
1.3.3 3. Prepare for a Pull Request	8
1.3.4 4. Submitting a Pull Request (PR) to the upstream repository . .	9
1.3.5 5. Cleaning Up	11
1.3.6 Workflow Summary	11
1.4 Branch Prefixes	12
1.5 Writing Commit Messages	12
1.5.1 Prefix your commit subject line with a tag	12
1.5.2 Formatting your message	12
1.5.3 Use the imperative mood in your commit subject line	15
1.5.4 Subject and body lines	15
2 Contributing To Web	16
2.1 Coding Guidelines	16

1 Doubtfire Git Workflow

We follow a Forking workflow¹ when developing Doubtfire.

1.1 Table of Contents

1. About the Doubtfire Branch Structure
2. Getting started with the Forking Workflow
3. Forking and Cloning the repository
4. Writing your new changes
5. Prepare for a Pull Request
6. Submitting a Pull Request (PR) to the upstream repository
7. Cleaning Up
8. Workflow Summary
9. Branch Prefixes
10. Writing Commit Messages
11. Prefix your commit subject line with a tag
12. Formatting your message
13. Use the imperative mood in your commit subject line
14. Subject and body lines

1.2 About the Doubtfire Branch Structure

We try to keep two main branches at all times:

- `master` for production
- `develop` for current development, a branch off `master`

That way, we follow the workflow:

1. branch off `develop`, giving your branch one of the prefixes defined below,
2. make your changes in that branch,
3. merge your branch back into `develop`,
4. delete your branch to clean up

¹See <https://www.atlassian.com/git/tutorials/comparing-workflows/forking-workflow>

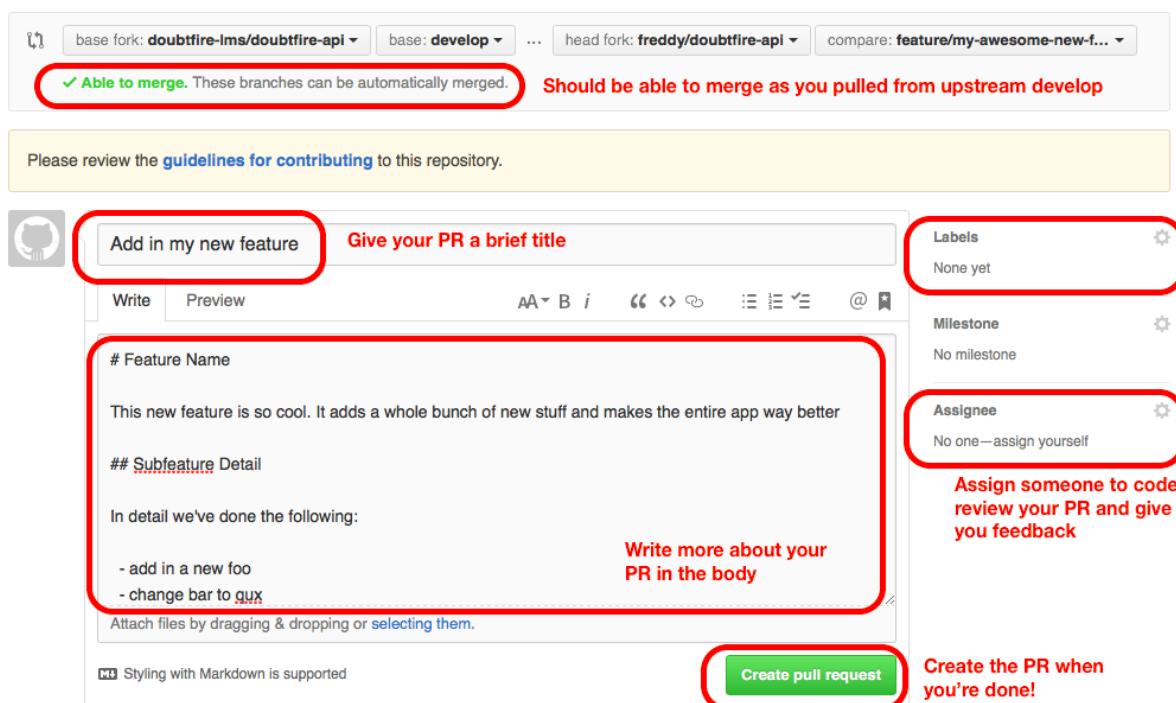


Figure 1: Feature Branches

In some cases, your branches may only consist of one or two commits. This is still okay as you can submit a pull request for code review back into `develop`.

You may want to branch again, e.g.:

```
* master
  \
  | \
  | |
  | (b1) develop
  | | \
  | | (b2) feature/my-new-feature
  | | | \
  | | | (b3) test/unit-tests-for-new-feature
  | | | /
  | | | (m1)
  | | /
  | | (m2)
  | | \
  | | (b4) fix/broken-thing
  | | /
  | (m3)
  | /|
  | /|
  (m4)
  | |
  | |
* *
```

Here, we:

1. branched off `master` to create our `develop` branch, at `b1`
2. branched off `develop` to create a new feature under the new branch `feature/my-new-feature`, at `b2`
3. branched off `feature/my-new-feature` to create some unit tests for that feature under `test/unit-tests-for-new-feature`, at `b3`

4. merged those unit tests back into `feature/my-new-feature`, at `m1`
5. merged the new feature back into `develop`, at `m2`
6. found a new bug in the feature later on, so branched off `develop` into `fix/broken-thing`, at `b4`
7. after we fixed our bug, we merged `fix/broken-thing` back into `develop`, at `m3`
8. decide we're ready to release, so merge `develop` into `master`, at `m4`

Note that along the way **we're deleting branches after we don't need them**. This helps us keep *short-lived* branches that don't go *stale* after months of inactivity, and prevents us from forgetting about open branches. The only branch we kept open was `develop`, which we can always branch off for new, un-released changes again.

Ideally, any changes that are merged into `master` have been **code-reviewed** before they were merged into `develop`. **You should always code review before merging back into develop**. You can do this by performing a Pull Request, where the reviewer can see the changes you want to merge in to `develop`.

1.3 Getting started with the Forking Workflow

1.3.1 1. Forking and Cloning the repository

1.3.1.1 Fork the Repo

To get a copy of a Doubtfire repositories on your user account, you will need to fork it *for each repository*:



Figure 2: Fork the repo

1.3.1.2 Clone the Fork

You can then clone the repositories you have forked to your machine. To do so, navigate to your forked repositories and copy the clone URL:

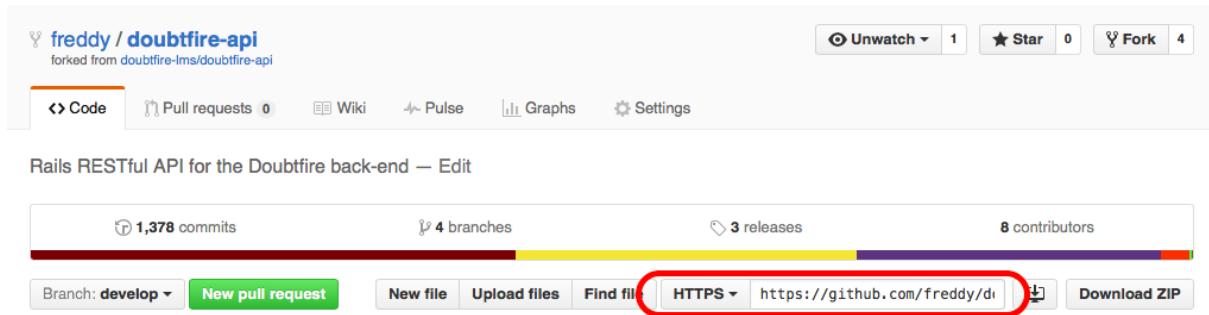


Figure 3: Copy the clone URL

Navigate to your `projects` or `repo` folder, and make a `doubtfire` folder. Then clone using the URLs you copied above:

```
$ cd ~/repos
$ mkdir doubtfire
$ cd doubtfire
$ git clone https://github.com/{username}/doubtfire-api.git
$ git clone https://github.com/{username}/doubtfire-web.git
```

1.3.1.3 Set up your upstream to `doubtfire-lms`

By default, git tracks your remote forked repository (the repository you cloned). This remote is called `origin`.

You will then need to set up a new remote to track to the `doubtfire-lms` owned repository. This will be useful when you need to get the latest changes other developers have contributed to the `doubtfire-lms` repo, but you do not yet have those changes in your forked repo. Call this remote `upstream`:

```
$ cd ~/repos/doubtfire/doubtfire-api
$ git remote add upstream https://github.com/doubtfire-lms/doubtfire-api.git
$ cd ~/repos/doubtfire/doubtfire-web
$ git remote add upstream https://github.com/doubtfire-lms/doubtfire-web.git
```

1.3.1.4 Ensure you have your author credentials set up

You should ensure your git user config are set and set to the email address you use with GitHub:

```
$ git config --global user.email "my-github-email@gmail.com"  
$ git config --global user.name "Freddy Smith"
```

1.3.1.5 Use a rebase pull

We also want to avoid having merge commits whenever you pull from `upstream`. It is useful to pull from upstream using the `--rebase` switch, as this avoids an unnecessary merge commit when pulling if there are conflicts.

To fix this, always pull with `--rebase` (unless otherwise specified—see the `--ff` switch needed in Step 3):

```
$ git pull upstream develop --rebase
```

or alternatively, make a rebase pull as your default setting:

```
$ git config --global pull.rebase true
```

1.3.2 2. Writing your new changes

As per the branching structure, you need to branch off of `develop` to a new branch that will have your code changes in it. When branching, **be sure you are using a branch prefix**:

```
$ cd ~/repos/doubtfire/doubtfire-api  
$ git checkout -b feature/my-awesome-new-feature
```

You can now begin making your changes. Commit along the way, **being sure to conform to the commit message guidelines**, on this branch and push to your fork:

```
$ git status
```

```
On branch feature/my-awesome-new-feature  
Your branch is up-to-date with 'origin/feature/my-awesome-new-feature'.  
Changes not staged for commit:  
(use "git add <file>..." to update what will be committed)  
(use "git checkout -- <file>..." to discard changes in working directory)
```

```
modified: src/file-that-changed.js
modified: src/another-file-that-changed.js
```

```
$ git add src/file-that-changed.js src/another-file-that-changed.js
$ git commit
```

```
[feature/my-awesome-new-feature 7f35016] DOCS: Add new documentation about git
 2 files changed, 10 insertions(+), 15 deletions(-)
```

```
$ git push -u origin feature/my-awesome-new-feature
```

Note you only need to add the `-u` flag on an initial commit for a new branch.

1.3.3 3. Prepare for a Pull Request

Note, while it is advised you perform this step, it you can skip it and move straight to the Pull Request step. If the branch cannot be automatically merged, then you should run through these steps.

When you are done with your changes, you need to pull any changes from `develop` from the `upstream` repository. This essentially means “get me anything that has changed on the `doubtfire-lms` repository that I don’t yet have”.

To do this, pull any changes (if any) from the `upstream` repository’s `develop` branch into your local `develop` branch:

```
$ git checkout feature/my-awesome-new-feature
$ git pull --ff upstream develop
```

If there are merge conflicts, you can resolve them now. Follow GitHub’s guide² for resolving merge conflicts.

We can now update your `origin` repository’s `my-awesome-new-feature` on GitHub such that it will include the changes from `upstream`:

```
$ git push origin feature/my-awesome-new-feature
```

²See <https://help.github.com/articles/resolving-a-merge-conflict-from-the-command-line>

1.3.4 4. Submitting a Pull Request (PR) to the upstream repository

Once you have pushed your changes to your fork, and have ensured nothing has broken, you can then submit a pull request for code review to Doubtfire.

To submit a pull request, go to the relevant Doubtfire LMS Repo and click “New Pull Request”:

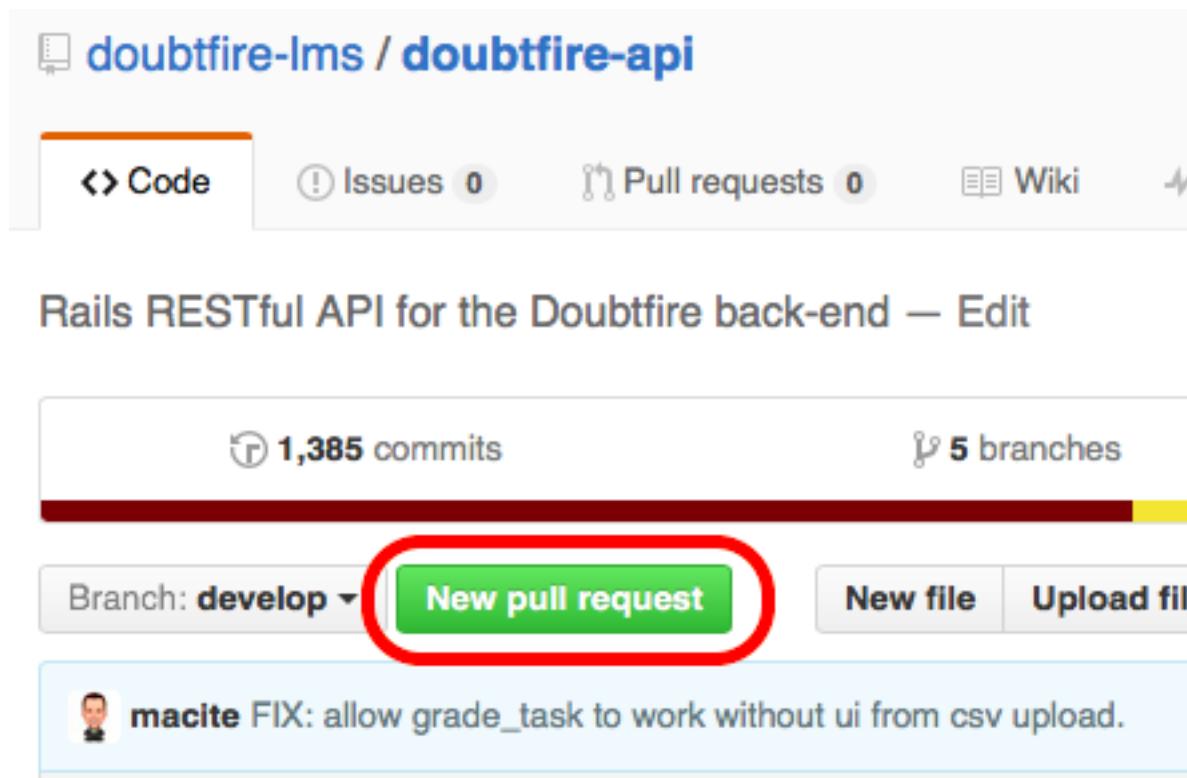


Figure 4: New PR

Ensure that the **Head Fork** is set to your forked repository and on your feature branch. If you cannot see your repository, try clicking the “Compare across forks” link.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

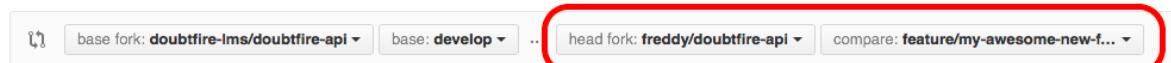


Figure 5: Compare forks

You can then begin writing the pull request. Be sure you are **Able to Merge**, otherwise **try repeating an upstream pull of develop into your feature branch, as per the previous step.**

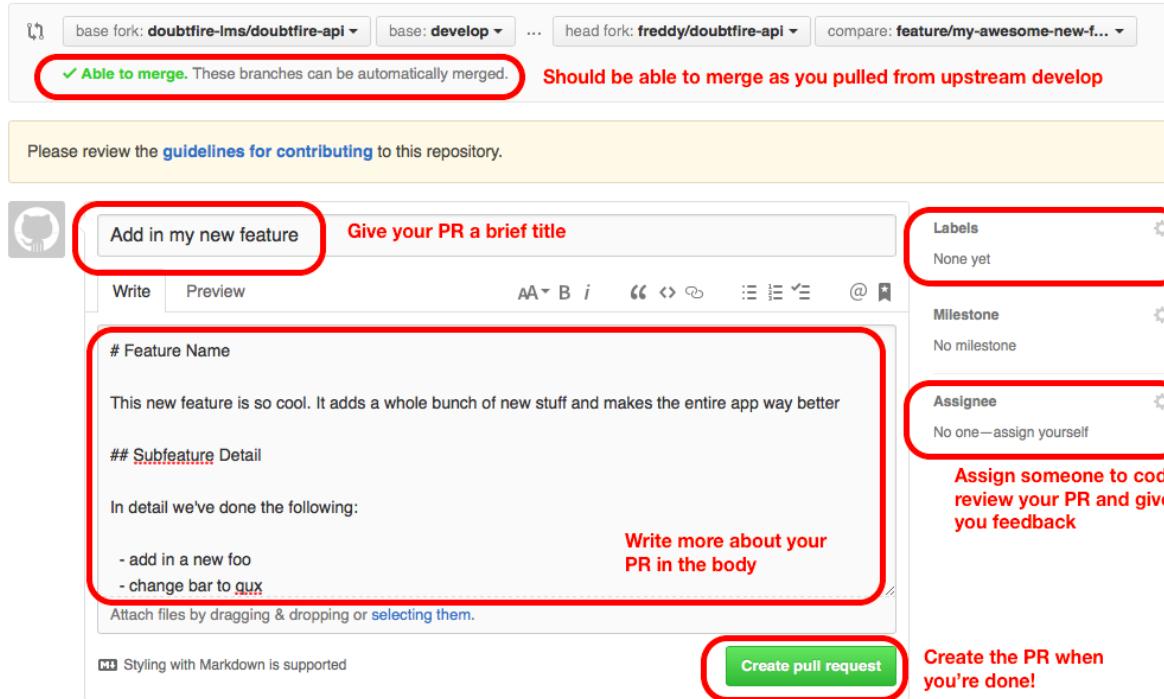


Figure 6: Writing a Pull Request

With your PR body, be descriptive. GitHub may automatically add a commit summary in the body. If fixing a problem, include a description of the problem you're trying to fix and why this PR fixes it. When you are done, assign a code reviewer and add a tag (if applicable) and create the pull request!

If your code is ok, it will be merged into `develop`, (and eventually `master`, meaning your code will go live - woohoo :tada:)

If not, the reviewer will give you suggestions and feedback for you to fix your code.

STOP! Continue to the next step once your Pull Request is approved and merged into the `doubtfire-lms`'s `develop` branch.

1.3.5 5. Cleaning Up

Once your pull request is approved, your code changes are finalised, and merged you will want to delete your old feature branch so you don't get lots of old branches on your repository.

Following from the example above, we would delete `feature/my-awesome-new-feature` as it has been merged into `develop`. We first delete the branch locally:

```
$ git branch -D feature/my-awesome-new-feature
```

Then remove it from your fork on GitHub:

```
$ git push origin --delete feature/my-awesome-new-feature
```

Then ensure you are git is no longer tracking the deleted branch from `origin` by running a fetch prune:

```
$ git fetch --prune
```

As your changes have been merged into `upstream`'s `develop` branch, pull from `upstream` and you can grab those changes into your local repository:

```
$ git checkout develop
$ git pull upstream develop
```

Then push those changes up into your `origin`'s `develop` so that it is synced with `upstream`'s `develop`:

```
$ git push origin upstream
```

1.3.6 Workflow Summary

Step 1. Set up for new feature branch:

```
$ git checkout develop          # make sure you are on develop
$ git pull --rebase upstream develop # sync your local develop with upstream's develop
$ git checkout -b my-new-branch    # create your new feature branch
```

Step 2. Make changes, and repeat until you are done:

```
$ git add ... ; git commit ; git push      # make changes, commit, and push to origin
```

Step 3. Submit a pull request, and if unable to merge:

```
$ git pull --ff upstream develop          # merge upstream's develop in your feature branch
$ git add ... ; git commit                # resolve merge conflicts and commit
$ git push origin                         # push your merge conflict resolution to origin
```

Step 4. Only when the pull request has been approved and merged, clean up:

```
$ git checkout develop                   # make sure you are back on develop
$ git branch -D my-new-branch           # delete the feature branch locally
$ git push --delete my-new-branch       # delete the feature branch on origin
$ git fetch origin --prune             # make sure you no longer track the deleted branch
$ git pull --rebase upstream develop    # pull the merged changes from develop
$ git push origin develop               # push to origin to sync origin with develop
```

1.4 Branch Prefixes

When branching, try to prefix your branch with one of the following prefixes shown in Table 1.

1.5 Writing Commit Messages

Parts of this section have been adapted from Chris Beam's post, How to Write Good Commit Messages³.

When writing commits, try to follow this guide as described in this subsection.

1.5.1 Prefix your commit subject line with a tag

Each one of your commit messages should be prefixed with one of the following shown in Table 2

1.5.2 Formatting your message

Capitalise your commit messages and do not end the subject line with a period

FIX: Change the behaviour of the logging system

³See <http://chris.beams.io/posts/git-commit/>

Table 1: Branch prefixes

Prefix	Description	Example
feature/	New feature was added	feature/add-learning-outcome-alignment
fix/	A bug was fixed	fix/crash-when-code-submission-finished
enhance/	Improvement to existing feature, but not visual enhancement (See LOOKS)	enhance/allow-code-files-to-be-submitted
looks/	UI Refinement, but not functional change (See ENHANCE)	looks/rebrand-ui-for-version-2-marketing
quality/	Refactoring of existing code	quality/make-code-convention-consistent
doc/	Documentation-related changes	doc/add-new-api-documentation
config/	Project configuration changes	config/add-framework-x-to-project
speed/	Performance-related improvements	speed/new-algorithm-to-process-foo
test/	Test addition or enhancement	test/unit-tests-for-new-feature-x

Table 2: Commit tagging guide

Tag	Description	Example
NEW	New feature was added	NEW: Add unit outcome alignment tab
FIX	A bug was fixed	FIX: Amend typo throwing error
ENHANCE	Improvement to existing feature, but not visual enhancement (See LOOKS)	ENHANCE: Calculate time between classes to show on timetable
LOOKS	UI Refinement, but not functional change (See ENHANCE)	LOOKS: Make plagiarism tab consistent with other tabs
QUALITY	Refactoring of existing code	QUALITY: Make directives in consistent format with each other
DOC	Documentation-related changes	DOC: Write guide on writing commit messages
CONFIG	Project configuration changes	CONFIG: Add new scheme for UI automation testing
SPEED	Performance-related improvements	SPEED: Reduce time needed to batch process PDF submissions
TEST	Test addition or enhancement	TEST: Add unit tests for tutorial administration

and not

fix: change the behaviour of the logging system.

1.5.3 Use the imperative mood in your commit subject line

Write your commits in the imperative mood and not the indicative mood

- “Fix a bug” and **not** “Fixed a bug”
- “Change the behaviour of Y” and **not** “*Changed* the behaviour of Y”
- “Add new API methods” and **not** “Sweet new API methods”

A properly formed git commit subject line should always be able to complete the following sentence:

If applied, this commit will **your subject line here**

If applied, this commit will **fix a bug**

If applied, this commit will **change the behaviour of Y**

and not

If applied, this commit will **sweet new API methods**

1.5.4 Subject and body lines

Write a commit subject, and explain that commit on a new line (if need be):

FIX: Derezz the master control program

MCP turned out to be evil and had become intent on world domination.
This commit throws Tron's disc into MCP (causing its deresolution)
and turns it back into a chess game.

Keep the subject line (top line) concise; keep it **within 50 characters**.

Use the body (lines after the top line) to explain why and what and *not* how; keep it **within 72 characters**.

1.5.4.1 But how can I write new lines if I'm using `git commit -m "Message"`?

Don't use the `-m` switch. Use a text editor to write your commit message instead.

If you are using the command line to write your commits, it is useful to set your git editor to make writing a commit body easier. You can use the following command to set your editor to `nano`, `emacs`, `vim`, `atom`.

```
$ git config --global core.editor nano
$ git config --global core.editor emacs
$ git config --global core.editor vim
$ git config --global core.editor "atom --wait"
```

If you want to use Sublime Text as your editor, follow this guide⁴.

If you are not using the command line for git, you probably should be⁵.

2 Contributing To Web

Please read through this document before contributing to Doubtfire.

Before continuing, **please read the contributing document⁶ of the API**, as this outlines the Git workflow you should be following.

2.1 Coding Guidelines

For extendability and maintenance purposes, following these guidelines:

- Name a directive with its role in mind (i.e., as a **Agent Noun⁷**) to give a small summary as to what the directive *does*:
- when *viewing* a project or task, the directive is **project-viewer** and **task-viewer**
- when *assessing* task submissions, the directive is **task-submission-assessor**
- when *editing* a unit's tutorials, the directive is **unit-tutorial-editor**

⁴See <https://help.github.com/articles/associating-text-editors-with-git/#using-sublime-text-as-your-editor>

⁵See <http://try.github.io>

⁶See <https://github.com/doubtfire-lms/doubtfire-api/blob/develop/CONTRIBUTING.md>

⁷See https://en.wikipedia.org/wiki/Agent_noun

- Name directives that show lots of data in one directive in a table a list: e.g.: `unit-student-list`, `group-member-list`, `project-top-task-list`
- Name directives with a series of steps to perform a goal a `wizard`, e.g.: `project-portfolio-wizard`, `new-user-wizard`
- Always name modals in Pascal Case `SomeModal` and create them as a factory/controller pair CoffeeScript file which can then be easily created on the fly:

```
# foo/modals/create-foo-modal.coffee
angular.module('doubfire.foo.modals.create-foo-modal', [])

#
# Prompts the user to create a Foo using a bar and qux variable
#
.factory('CreateFooModal', ($modal) ->
  CreateFooModal = {}

CreateFooModal.show = (bar, qux) ->
  $modal.open
    templateUrl: 'foo/modals/create-foo-modal.tpl.html'
    controller: 'CreateFooModalCtrl'
    resolve:
      bar: -> bar
      qux: -> qux

CreateFooModal
)

.controller('CreateFooModalCtrl', ($scope, bar, qux) ->
  # Does stuff with bar and qux to create a foo
  $scope.bar = bar
  $scope.qux = qux
)
# foo/states/foo-view/foo-view.coffee
```

```
# ...
.controller('FooViewCtrl', ($scope, CreateFooModal) ->
# ...
$createNewFoo = ->
  CreateFooModal.show($scope.bar, $scope.qux)
)
```

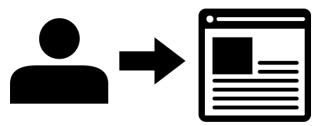
- Always name non-anonymous controllers with a `Ctrl` suffix
- Case correctly:
- `directiveName` should be camelCase - refer to this Angular documentation⁸
- `ServiceName, ControllerNameCtrl` should be in PascalCase
- Regardless of abbreviations, stick to these conventions (e.g., `pdfPanelViewer` directive works, but `PDFPanelViewer` won't work as it needs to be camelCase)
- Place modals and states in a `modals` and `states` folder under the root. All else can be in their own folders unless they are of a related concept (see the `project-portfolio-wizard` folder under `project`, `stats` under `tasks` and `units`)
- The name of a module should follow the directory structure of where it has been placed (i.e., in the above example, the template file was at `foo/modals/create-foo-modal.tpl.html`, the CoffeeScript file was at `foo/modals/create-foo-modal.coffee`, and thus the module is `doubtfire.foo.modals.create`)
- Try to give a brief summary of what the directive, state or factory does. E.g., the comment in the example above for `CreateFooModal` is sufficient.
- Try to abstract as much code inside a model class as possible. At present a lot of this code is in a model's service, and it should be moved into the model's resource definition as much as possible:

`Unit.addTutorial tutorialData`

instead of:

`unitService.addTutorial unit, tutorialData`

⁸See <https://docs.angularjs.org/guide/directive#normalization>



Chapter 10

User Manual

Contents

1 Helpdesk Ticketing System User Manual	2
1.1 For students	2
1.1.1 Creating tickets	2
1.1.2 Closing a ticket	5
1.2 For staff	8
1.2.1 Clocking on	8
1.2.2 Clocking off	10

1 Helpdesk Ticketing System User Manual

All helpdesk-related functionality items are located in the main navigation bar underneath the *Helpdesk* menu. Refer to Figure 1.

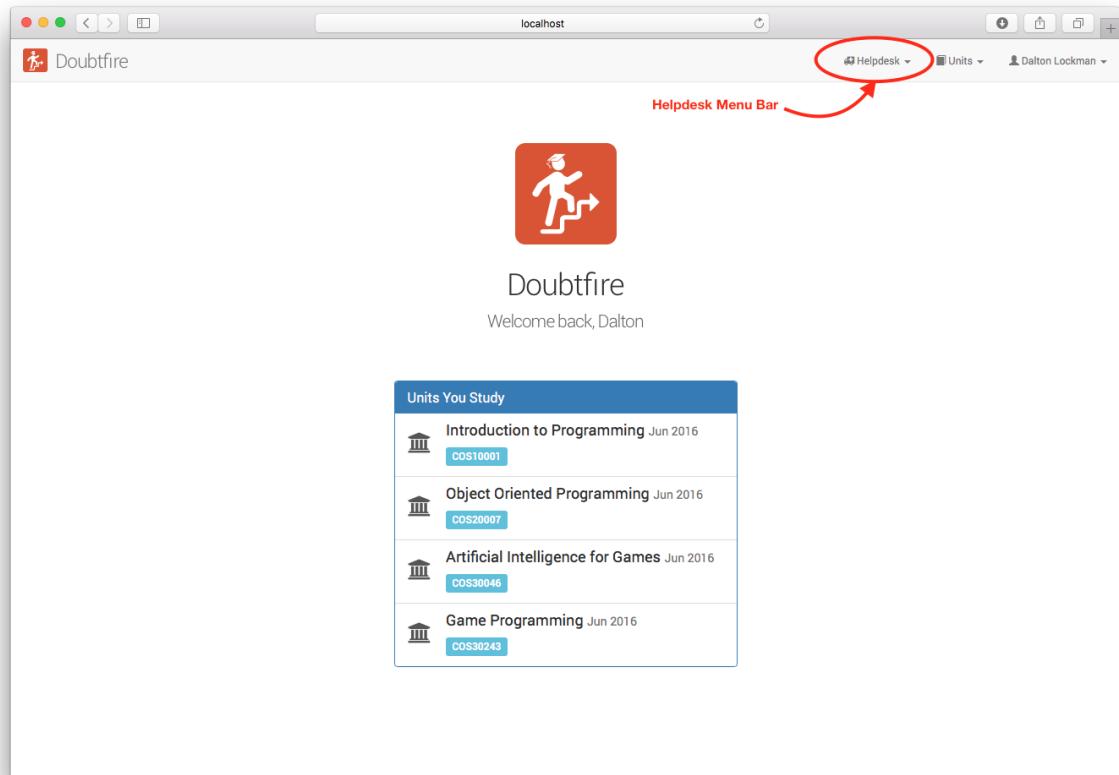


Figure 1: Helpdesk menu

Clicking on this will reveal all things you can do either as a student or as a staff member.

1.1 For students

1.1.1 Creating tickets

Creating tickets is the way in which you can request a tutor to get your assistance. No longer do you need to raise your hand up. Doubtfire can do this all for you!

Underneath the helpdesk menubar, click on the *Submit a Ticket* link as illustrated in Figure 2.

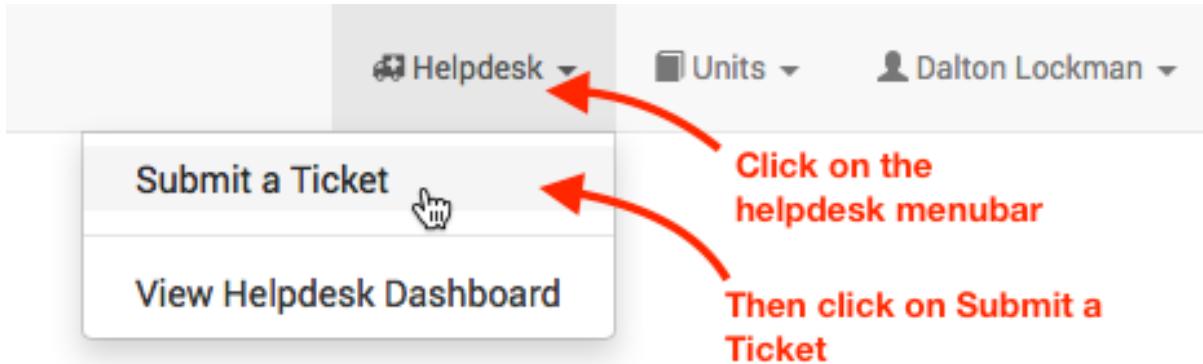


Figure 2: Submit ticket link from the helpdesk menu

This will present you with a modal dialog menu from which you can enter in:

1. the unit you would like assistance with,
2. the task you would like assistance with, and
3. any additional comments you would like help with.

This is shown in Figure 3 below.

Submit a Ticket

Having trouble with something? Get help from a tutor at the programming helpdesk, located at ATC620.

Unit you
need help
with

Unit

COS10001

Introduction to Programming ▾

Task

Select Task ▾

Task you need help with

Description

Enter description



Ticket
description

Give the tutor some context to your issue by providing a description of your question.

Submits the ticket

Close

Submit Ticket

You can choose to leave items 2 and 3 out as optional. For example, if you only need help with setting up your laptop with the unit's required software, you can skip the task and simply add some comments such as that displayed in the screenshot below.

Simply select the unit you would like help with from the dropdown and the task also, if applicable. The description can accept Markdown-formatted text, so you can even submit code snippets if you want. Refer to the illustration in Figure 4. You can preview your description by clicking the eye icon, or if you need help with Markdown formatting, click on the question-mark.



Figure 3: You can add Markdown-formatted code to your description

Once you submit your ticket, as illustrated by the green button in Figure 3, the navigation bar will display an orange ticket symbol indicating that you have a helpdesk ticket open.

Hovering over the ticket will show how long the ticket has been open for (Figure 5), and clicking it will show the *View Open Ticket* modal. Refer to the following section for more details about this modal.

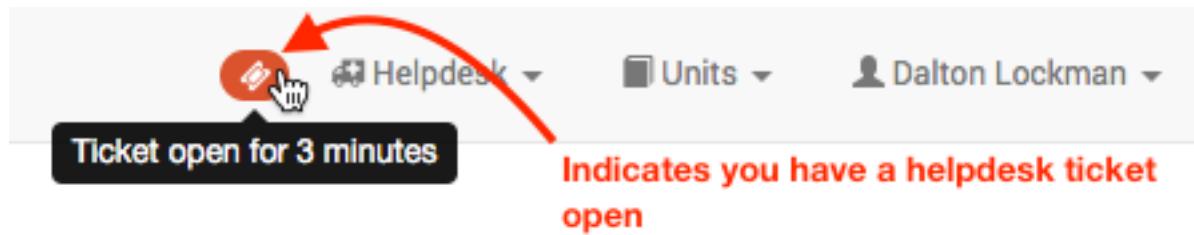


Figure 4: Hovering over a ticket will indicate how long that ticket has been open for

1.1.2 Closing a ticket

If you need to leave the helpdesk or no longer need assistance, you should close your ticket so that a tutor doesn't come around to you to help you when they don't need to. Follow the steps below to close a ticket.

Underneath the helpdesk menubar, click on the *View Open Ticket* link as illustrated in Figure 6.

Currently Open Ticket

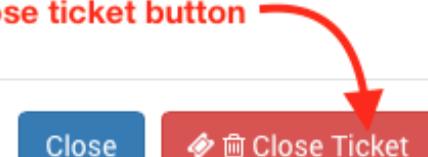
Unit	COS10001	Introduction to Programming
Task	A1	Assignment 1
Description	I keep getting this compiler error: <code>clang: error: no input files</code> . I added a file?	
In Queue For	8 minutes (1:28 PM)	
		Close ticket button
		 A red arrow points from the text "Close ticket button" to the "Close Ticket" button below.
		Close  Close Ticket

Figure 5: Opening the View Open Ticket link

The *Currently Open Ticket* modal should appear (Figure 7). From here, you can:

- view the details of your ticket which you entered when you created it,
- view how long you have been in queue for, and
- close your ticket.

When you click the close ticket button, a confirmation message will appear that confirms if you want to close your ticket. Note that if you do, you will have to rejoin the queue from the back of the queue if you want to make a new ticket.

Currently Open Ticket

Unit	COS10001 Introduction to Programming
Task	A1 Assignment 1
Description	I keep getting this compiler error: <code>clang: error: no input files</code> . I added a file?
In Queue For	8 minutes (1:28 PM)
Close ticket button	
	 A blue rectangular button labeled "Close". To its right is a red rectangular button labeled "Close Ticket" with a small trash can icon next to the word "Close". A red curved arrow points from the text "Close ticket button" above to the red "Close Ticket" button.

Figure 6: Currently Open Ticket Modal

1.2 For staff

1.2.1 Clocking on

When you are ready to start working at the helpdesk, navigate to the helpdesk main menubar, click on it and then click on the *Clock On* link. This is shown in Figure 8.

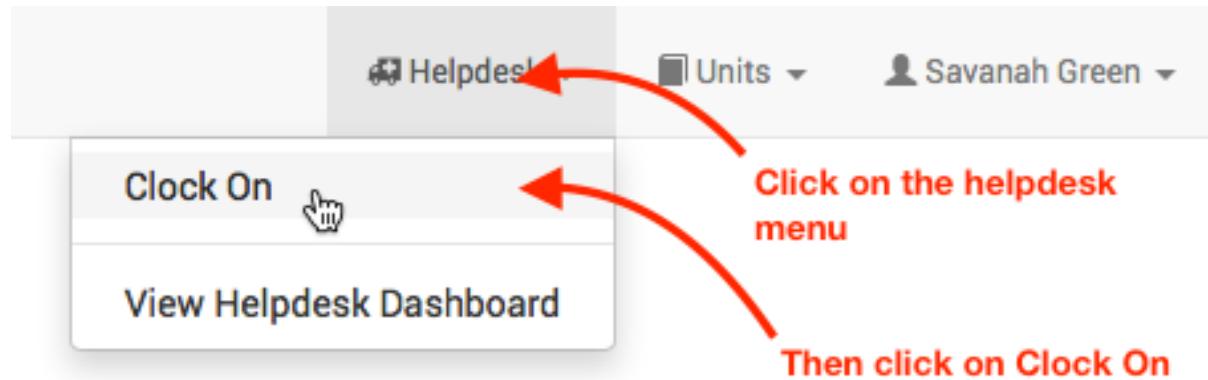


Figure 7: Clock On Link

The *Clock On* modal will then appear. Simply enter in the number of hours you are working for in decimal format and click the green clock on button. The helpdesk ticketing system will automatically clock you off after the work time you have entered in.

In the example shown in Figure 9, the user has entered 3 hours and 15 minutes. This is because 0.25 in hours is 15 minutes. There are some restrictions on this entry such as:

1. working more than 8 consecutive hours is not allowed,
2. you must work for more than 15 minutes, and
3. you can't enter in invalid data such as non-numeric characters.

Clicking the clock on button will clock you on, and this is indicated with the orange clock icon shown next to the helpdesk. You can see how long you have left in your shift by hovering over this icon (Figure 10). Clicking this icon will display the current shift modal (see the following section for more).

During this time, you can resolve tickets by viewing the dashboard.

Clock on to the helpdesk

Please enter how long you are working at the helpdesk for.

 hours

You will be automatically clocked off at 05:03 pm, but you can manually clock off at an earlier time.

[Close](#)

Figure 8: Clocking onto the helpdesk requires you to enter in your work hours

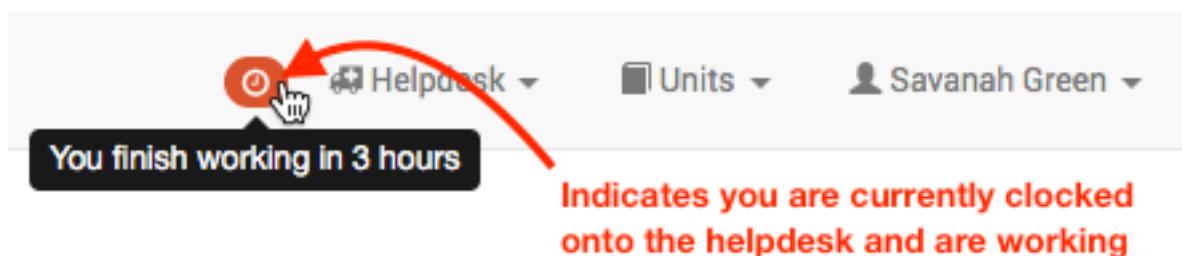
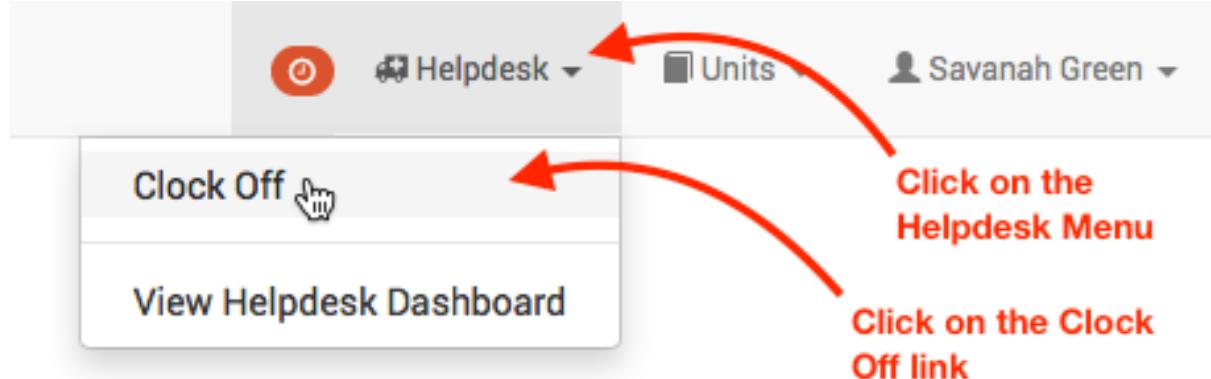


Figure 9: Hovering over the shift icon tells you when the helpdesk will clock you off automatically

1.2.2 Clocking off

If you need to leave the helpdesk immediately before the end of your shift, you can prematurely clock off.

To do so, click on the helpdesk menubar and then the clock off link as shown in Figure 11.



The *Clock Off* modal should then appear. From here you can see how long you have left in your helpdesk shift. To clock off prematurely, click the red clock off button as shown in Figure 12.

Clock off from the helpdesk

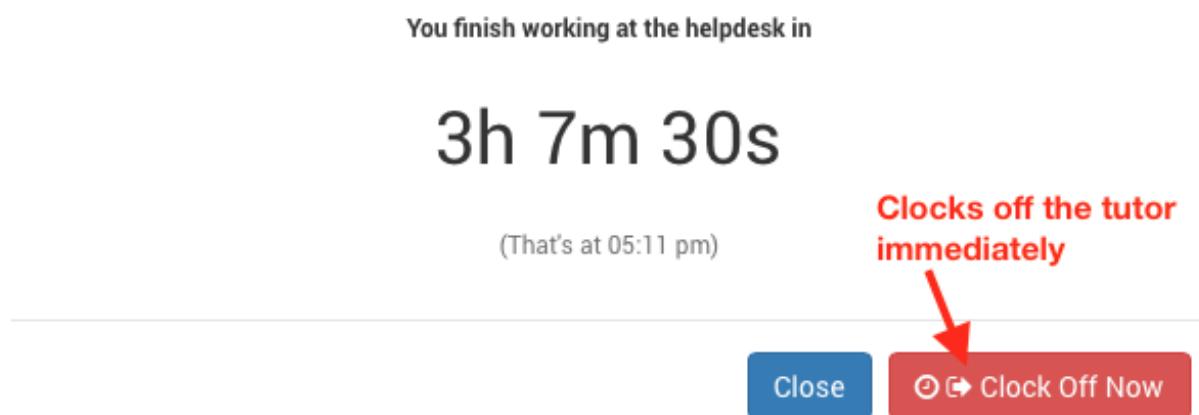
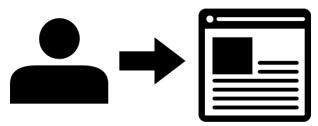


Figure 10: Clock off immediately

Clicking this button will warn you with a confirmation modal. Confirming on the confirmation modal will clock you off from the helpdesk immediately before your original automatic clock off time.



Chapter 11

Source Code Directory Dump

Contents

1 Doubtfire API	2
2 Doubtfire Web	6

1 Doubtfire API

```
doubtfire-api/app
|-- api
|   |-- api.rb
|   |-- auth.rb
|   |-- group_sets.rb
|   |-- helpdesk
|       |-- session.rb
|       |-- stats.rb
|       \-- ticket.rb
|   |-- learning_alignment.rb
|   |-- learning_outcomes.rb
|   |-- projects.rb
|   |-- students.rb
|   |-- submission
|       |-- batch_task.rb
|       |-- generate.rb
|       |-- generate_helpers.rb
|       |-- portfolio_api.rb
|       \-- portfolio_evidence_api.rb
|   |-- task_comments.rb
|   |-- task_definitions.rb
|   |-- tasks.rb
|   |-- tutorials.rb
|   |-- unit_roles.rb
|   |-- units.rb
|   \-- users.rb
|-- controllers
|   |-- application_controller.rb
|   |-- lecture_resource_downloads_controller.rb
|   \-- portfolio_downloads_controller.rb
|-- helpers
|   |-- application_helper.rb
```

```
|   |-- auth_helpers.rb
|   |-- authorisation_helpers.rb
|   |-- csv_helper.rb
|   |-- db_helpers.rb
|   |-- doubtfire_logger.rb
|   |-- file_helper.rb
|   |-- log_helper.rb
|   |-- mime-check-helpers.rb
|   \-- timeout_helper.rb
|-- mailers
|   |-- convenor_contact_mailer.rb
|   \-- portfolio_evidence_mailer.rb
|-- models
|   |-- badge.rb
|   |-- group.rb
|   |-- group_membership.rb
|   |-- group_set.rb
|   |-- group_submission.rb
|   |-- helpdesk_session.rb
|   |-- helpdesk_ticket.rb
|   |-- learning_outcome.rb
|   |-- learning_outcome_task_link.rb
|   |-- login.rb
|   |-- plagiarism_match_link.rb
|   |-- portfolio_evidence.rb
|   |-- progress.rb
|   |-- project.rb
|   |-- role.rb
|   |-- sub_task.rb
|   |-- sub_task_definition.rb
|   |-- task.rb
|   |-- task_comment.rb
|   |-- task_definition.rb
```

```
|   |-- task_engagement.rb
|   |-- task_status.rb
|   |-- task_submission.rb
|   |-- tutorial.rb
|   |-- unit.rb
|   |-- unit_role.rb
|   \-- user.rb
|-- serializers
|   |-- group_serializer.rb
|   |-- group_set_serializer.rb
|   |-- helpdesk_session_serializer.rb
|   |-- helpdesk_ticket_serializer.rb
|   |-- learning_outcome_serializer.rb
|   |-- learning_outcome_task_link_serializer.rb
|   |-- project_serializer.rb
|   |-- role_serializer.rb
|   |-- task_comment_serializer.rb
|   |-- task_definition_serializer.rb
|   |-- task_serializer.rb
|   |-- tutorial_serializer.rb
|   |-- unit_role_serializer.rb
|   |-- unit_serializer.rb
|   |-- user_role_serializer.rb
|   \-- user_serializer.rb
\-- views
  |-- layouts
  |   \-- application.pdf.erbtex
  |-- portfolio
  |   \-- portfolio_pdf.pdf.erb
  |-- portfolio_evidence_mailer
  |   |-- portfolio_failed.html.erb
  |   |-- portfolio_failed.text.erb
  |   \-- portfolio_ready.html.erb
```

```
|   |-- portfolio_ready.text.erb
|   |-- task_feedback_ready.html.erb
|   |-- task_feedback_ready.text.erb
|   |-- task_pdf_failed.html.erb
|   |-- task_pdf_failed.text.erb
|   |-- task_pdf_ready_message.html.erb
|   \-- task_pdf_ready_message.text.erb
|-- shared
|   \-- _file.pdf.erb
\-- task
    \-- task_pdf.pdf.erb
doubtfire-api/db
|-- migrate
|   |-- 20160705025331_create_helpdesk_tickets.rb
|   |-- 20160708011422_change_helpdesk_ticket_comment_to_description.rb
|   |-- 20160722114921_create_helpdesk_sessions.rb
|   |-- 20160725141011_add_resolved_at_to_helpdesk_ticket.rb
|   |-- 20160725144833_add_minutes_to_resolve_to_helpdesk_ticket.rb
|   \-- 20160826233551_add_is_closed_to_helpdesk_ticket.rb
|-- schema.rb
\-- seeds.rb
doubtfire-api/lib
|-- assets
|-- helpers
|   |-- database_populator.rb
|   |-- find_or_create_students.rb
|   \-- randomizer.rb
|-- shell
|   \-- timeout.sh
|-- tasks
|   |-- checks.rake
|   |-- compress_pdfs.rake
|   \-- demo.rake
```

```
|   |-- generate_pdfs.rake
|   |-- init.rake
|   |-- populate.rake
|   |-- test_setup.rake
|   |-- update_progress.rake
|   \-- update_temporal.rake
\-- templates
    \-- erb
        \-- scaffold
            \-- _form.html.erb
doubtfire-api/test
|-- api
|   |-- auth_test.rb
|   |-- helpdesk
|   |   \-- ticket_test.rb
|   \-- units_test.rb
|-- helpers
|   |-- auth_helper.rb
|   \-- json_helper.rb
|-- models
|   |-- helpdesk
|   |   \-- ticket_test.rb
|   \-- user_test.rb
\-- test_helper.rb
```

27 directories, 212 files

2 Doubtfire Web

```
doubtfire-web/src
|-- app
|   |-- api
|   |   |-- api-url.coffee
```

```
|   |   |-- api.coffee
|   |   |-- models
|   |   |   |-- convenor.coffee
|   |   |   |-- group-member.coffee
|   |   |   |-- group-set.coffee
|   |   |   |-- group.coffee
|   |   |   |-- helpdesk-session.coffee
|   |   |   |-- helpdesk-stats.coffee
|   |   |   |-- helpdesk-ticket.coffee
|   |   |   |-- intended-learning-outcome.coffee
|   |   |   |-- learning-alignments.coffee
|   |   |   |-- models.coffee
|   |   |   |-- portfolion-submission.coffee
|   |   |   |-- project.coffee
|   |   |   |-- students.coffee
|   |   |   |-- task-alignment.coffee
|   |   |   |-- task-comment.coffee
|   |   |   |-- task-completion-csv.coffee
|   |   |   |-- task-definition.coffee
|   |   |   |-- task-feedback.coffee
|   |   |   |-- task-similarity.coffee
|   |   |   |-- task.coffee
|   |   |   |-- tutor.coffee
|   |   |   |-- tutorial.coffee
|   |   |   |-- unit-role.coffee
|   |   |   |-- unit.coffee
|   |   |   |-- user-role.coffee
|   |   |   \-- user.coffee
|   |   \-- resource-plus.coffee
|   |-- app.coffee
|   |-- common
|   |   |-- alert-list
|   |   |   |-- alert-list.coffee
```

```
|   |   |   |-- alert-list.tpl.html
|   |   |-- common.coffee
|   |   |-- file-uploader
|   |   |   |-- file-uploader.coffee
|   |   |   |-- file-uploader.scss
|   |   |   |-- file-uploader.tpl.html
|   |   |-- filters
|   |   |   |-- filters.coffee
|   |   |-- grade-icon
|   |   |   |-- grade-icon.coffee
|   |   |   |-- grade-icon.scss
|   |   |   |-- grade-icon.tpl.html
|   |   |-- header
|   |   |   |-- header.coffee
|   |   |   |-- header.scss
|   |   |   |-- header.tpl.html
|   |   |-- helpdesk-header
|   |   |   |-- helpdesk-header.coffee
|   |   |   |-- helpdesk-header.scss
|   |   |   |-- helpdesk-header.tpl.html
|   |   |-- markdown-editor
|   |   |   |-- markdown-editor.coffee
|   |   |   |-- markdown-editor.scss
|   |   |   |-- markdown-editor.tpl.html
|   |   |-- modals
|   |   |   |-- about-doubtfire-modal
|   |   |   |   |-- about-doubtfire-modal.coffee
|   |   |   |   |-- about-doubtfire-modal.scss
|   |   |   |   |-- about-doubtfire-modal.tpl.html
|   |   |   |-- confirmation-modal
|   |   |   |   |-- confirmation-modal.coffee
|   |   |   |   |-- confirmation-modal.scss
|   |   |   |   |-- confirmation-modal.tpl.html
```

```
|   |   |   |-- csv-result-modal
|   |   |   |   |-- csv-result-modal.coffee
|   |   |   |   |-- csv-result-modal.scss
|   |   |   |   \-- csv-result-modal.tpl.html
|   |   |   |-- modals.coffee
|   |   |   \-- progress-modal
|   |   |       |-- progress-modal.coffee
|   |   |       \-- progress-modal.tpl.html
|   |   |-- pdf-panel-viewer
|   |   |   |-- pdf-panel-viewer.coffee
|   |   |   |-- pdf-panel-viewer.scss
|   |   |   \-- pdf-panel-viewer.tpl.html
|   |   |-- services
|   |   |   |-- alert-service.coffee
|   |   |   |-- analytics-service.coffee
|   |   |   |-- date-service.coffee
|   |   |   |-- grade-service.coffee
|   |   |   |-- group-service.coffee
|   |   |   |-- header-service.coffee
|   |   |   |-- outcome-service.coffee
|   |   |   |-- project-service.coffee
|   |   |   |-- redirect-service.coffee
|   |   |   |-- services.coffee
|   |   |   |-- task-service.coffee
|   |   |   \-- unit-service.coffee
|   |   \-- status-icon
|       |-- status-icon.coffee
|       |-- status-icon.scss
|       \-- status-icon.tpl.html
|   |-- config
|   |   |-- analytics
|   |   |   \-- analytics.coffee
|   |   |-- config.coffee
```

```
|   |   |-- contributors
|   |   |   |-- contributors.coffee
|   |   |-- debug
|   |   |   |-- debug.coffee
|   |   |-- local-storage
|   |   |   |-- local-storage.coffee
|   |   |-- root-controller
|   |   |   |-- root-controller.coffee
|   |   |-- routing
|   |   |   |-- routing.coffee
|   |   |-- runtime
|   |   |   |-- runtime.coffee
|   |   |-- vendor-dependencies
|   |   |   |-- vendor-dependencies.coffee
|   |-- errors
|   |   |-- errors.coffee
|   |   |-- errors.scss
|   |-- states
|   |   |-- not-found
|   |   |   |-- not-found.coffee
|   |   |   |-- not-found.tpl.html
|   |   |-- states.coffee
|   |   |-- timeout
|   |   |   |-- timeout.coffee
|   |   |   |-- timeout.tpl.html
|   |   |-- unauthorised
|   |   |   |-- unauthorised.coffee
|   |   |   |-- unauthorised.tpl.html
|   |-- groups
|   |   |-- group-member-contribution-assigner
|   |   |   |-- group-member-contribution-assigner.coffee
|   |   |   |-- group-member-contribution-assigner.scss
|   |   |   |-- group-member-contribution-assigner.tpl.html
```

```
|   |   |-- group-member-list
|   |   |   |-- group-member-list.coffee
|   |   |   |-- group-member-list.scss
|   |   |   \-- group-member-list.tpl.html
|   |   |-- group-selector
|   |   |   |-- group-selector.coffee
|   |   |   |-- group-selector.scss
|   |   |   \-- group-selector.tpl.html
|   |   |-- groups.coffee
|   |   |-- groupset-group-manager
|   |   |   |-- groupset-group-manager.coffee
|   |   |   |-- groupset-group-manager.scss
|   |   |   \-- groupset-group-manager.tpl.html
|   |   |-- groupset-selector
|   |   |   |-- groupset-selector.coffee
|   |   |   |-- groupset-selector.scss
|   |   |   \-- groupset-selector.tpl.html
|   |   |-- student-group-manager
|   |   |   |-- student-group-manager.coffee
|   |   |   \-- student-group-manager.tpl.html
|   |   \-- tutor-group-manager
|       |   |-- tutor-group-manager.coffee
|       |   \-- tutor-group-manager.tpl.html
|   |-- helpdesk
|   |   |-- helpdesk-ticket
|   |   |   |-- helpdesk-ticket.coffee
|   |   |   |-- helpdesk-ticket.scss
|   |   |   \-- helpdesk-ticket.tpl.html
|   |   |-- helpdesk.coffee
|   |   |-- modals
|   |   |   |-- modals.coffee
|   |   |   |-- session-modal
|   |   |   |   |-- session-modal.coffee
```

```
|   |   |   |   |-- session-modal.scss
|   |   |   |   \-- session-modal.tpl.html
|   |   |   \-- ticket-modal
|   |   |       |-- ticket-modal.coffee
|   |   |       |-- ticket-modal.scss
|   |   |       \-- ticket-modal.tpl.html
|   |   \-- states
|   |       |-- dashboard
|   |       |   |-- dashboard.coffee
|   |       |   |-- dashboard.scss
|   |       |   \-- dashboard.tpl.html
|   |       \-- states.coffee
|   |-- home
|   |   |-- home.coffee
|   |   \-- states
|   |       |-- home
|   |       |   |-- home.coffee
|   |       |   |-- home.scss
|   |       |   \-- home.tpl.html
|   |       \-- new-user-wizard
|   |           |-- new-user-wizard.coffee
|   |           |-- new-user-wizard.scss
|   |           \-- new-user-wizard.tpl.html
|   |       \-- states.coffee
|   |-- projects
|   |   |-- project-lab-list
|   |   |   |-- project-lab-list.coffee
|   |   |   \-- project-lab-list.tpl.html
|   |   \-- project-outcome-alignment
|   |       |-- project-outcome-alignment.coffee
|   |       \-- project-outcome-alignment.tpl.html
|   |   \-- project-portfolio-wizard
|   |       |-- portfolio-add-extra-files-step
```

```
| | | | | -- portfolio-add-extra-files-step.coffee
| | | | | -- portfolio-add-extra-files-step.scss
| | | | \-- portfolio-add-extra-files-step.tpl.html
| | | | |-- portfolio-grade-select-step
| | | | | -- portfolio-grade-select-step.coffee
| | | | | -- portfolio-grade-select-step.scss
| | | | \-- portfolio-grade-select-step.tpl.html
| | | | |-- portfolio-learning-summary-report-step
| | | | | -- portfolio-learning-summary-report-step.coffee
| | | | | -- portfolio-learning-summary-report-step.tpl.html
| | | | |-- portfolio-review-step
| | | | | -- portfolio-review-step.coffee
| | | | | -- portfolio-review-step.scss
| | | | \-- portfolio-review-step.tpl.html
| | | | |-- portfolio-tasks-step
| | | | | -- portfolio-tasks-step.coffee
| | | | | -- portfolio-tasks-step.scss
| | | | \-- portfolio-tasks-step.tpl.html
| | | | |-- portfolio-welcome-step
| | | | | -- portfolio-welcome-step.coffee
| | | | | -- portfolio-welcome-step.tpl.html
| | | | |-- project-portfolio-wizard.coffee
| | | | | -- project-portfolio-wizard.scss
| | | | \-- project-portfolio-wizard.tpl.html
| | | | |-- project-progress-dashboard
| | | | | -- project-progress-dashboard.coffee
| | | | \-- project-progress-dashboard.tpl.html
| | | | |-- project-viewer
| | | | | -- project-viewer.coffee
| | | | \-- project-viewer.tpl.html
| | | | |-- projects.coffee
| | | | \-- states
| | | | | -- feedback
```

```
|   |       |   |-- feedback.coffee
|   |       |-- show
|   |       |   |-- show.coffee
|   |       |   |-- show.scss
|   |       |   |-- show.tpl.html
|   |       |-- states.coffee
|
|   |-- sessions
|   |   |-- auth
|   |   |   |-- auth.coffee
|   |   |   |-- http-auth-injector.coffee
|   |   |   |-- roles
|   |   |   |   |-- if-role.coffee
|   |   |   |   |-- roles.coffee
|   |   |-- cookies
|   |   |   |-- cookies.coffee
|   |   |-- current-user
|   |   |   |-- current-user.coffee
|   |   |-- sessions.coffee
|   |   |-- sessions.spec.coffee
|   |   |-- states
|   |       |-- sign-in
|   |       |   |-- sign-in.coffee
|   |       |   |-- sign-in.scss
|   |       |   |-- sign-in.tpl.html
|   |       |-- sign-out
|   |       |   |-- sign-out.coffee
|   |       |   |-- sign-out.tpl.html
|   |       |-- states.coffee
|
|   |-- tasks
|   |   |-- modals
|   |   |   |-- grade-task-modal
|   |   |   |   |-- grade-task-modal.coffee
|   |   |   |   |-- grade-task-modal.scss
```

```
|   |   |   |   \-- grade-task-modal.tpl.html
|   |   |   \-- modals.coffee
|   |   |-- project-tasks-list
|   |   |   |-- project-tasks-list.coffee
|   |   |   |-- project-tasks-list.scss
|   |   |   \-- project-tasks-list.tpl.html
|   |   |-- project-top-tasks-list
|   |   |   |-- project-top-tasks-list.coffee
|   |   |   |-- project-top-tasks-list.scss
|   |   |   \-- project-top-tasks-list.tpl.html
|   |   |-- stats
|   |   |   |-- stats.coffee
|   |   |   |-- task-completion-stats
|   |   |   |   |-- task-completion-stats.coffee
|   |   |   |   \-- task-completion-stats.tpl.html
|   |   |   |-- task-status-stats
|   |   |   |   |-- task-status-stats.coffee
|   |   |   |   \-- task-status-stats.tpl.html
|   |   |   \-- task-summary-stats
|   |   |       |-- task-summary-stats.coffee
|   |   |       \-- task-summary-stats.tpl.html
|   |   |-- task-comments-viewer
|   |   |   |-- task-comments-viewer.coffee
|   |   |   |-- task-comments-viewer.scss
|   |   |   \-- task-comments-viewer.tpl.html
|   |   |-- task-definition-editor
|   |   |   |-- task-definition-editor.coffee
|   |   |   |-- task-definition-editor.scss
|   |   |   \-- task-definition-editor.tpl.html
|   |   |-- task-definition-selector
|   |   |   |-- task-definition-selector.coffee
|   |   |   |-- task-definition-selector.scss
|   |   |   \-- task-definition-selector.tpl.html
```

```
|   |   |-- task-feedback-assessor
|   |   |   |-- task-feedback-assessor.coffee
|   |   |   \-- task-feedback-assessor.tpl.html
|   |   |-- task-feedback-assessor-list
|   |   |   |-- task-feedback-assessor-list.coffee
|   |   |   \-- task-feedback-assessor-list.tpl.html
|   |   |-- task-ilo-alignment
|   |   |   |-- modals
|   |   |   |   |-- task-ilo-alignment-modal
|   |   |   |   |   |-- task-ilo-alignment-modal.coffee
|   |   |   |   |   \-- task-ilo-alignment-modal.tpl.html
|   |   |   |   \-- task-ilo-alignment.coffee
|   |   |   |-- task-ilo-alignment-editor
|   |   |   |   |-- task-ilo-alignment-editor.coffee
|   |   |   |   |-- task-ilo-alignment-editor.scss
|   |   |   |   \-- task-ilo-alignment-editor.tpl.html
|   |   |   |-- task-ilo-alignment-rater
|   |   |   |   |-- task-ilo-alignment-rater.coffee
|   |   |   |   |-- task-ilo-alignment-rater.scss
|   |   |   |   \-- task-ilo-alignment-rater.tpl.html
|   |   |   |-- task-ilo-alignment-viewer
|   |   |   |   |-- task-ilo-alignment-viewer.coffee
|   |   |   |   |-- task-ilo-alignment-viewer.scss
|   |   |   |   \-- task-ilo-alignment-viewer.tpl.html
|   |   |   \-- task-ilo-alignment.coffee
|   |   |-- task-plagiarism-file-viewer
|   |   |   |-- task-plagiarism-file-viewer.coffee
|   |   |   |-- task-plagiarism-file-viewer.scss
|   |   |   \-- task-plagiarism-file-viewer.tpl.html
|   |   |-- task-plagiarism-report-viewer
|   |   |   |-- task-plagiarism-report-viewer.coffee
|   |   |   \-- task-plagiarism-report-viewer.tpl.html
|   |   |-- task-sheet-viewer
```

```
|   |   |   |-- task-sheet-viewer.coffee
|   |   |   |-- task-sheet-viewer.scss
|   |   |   \-- task-sheet-viewer.tpl.html
|   |   |-- task-status-selector
|   |   |   |-- task-status-selector.coffee
|   |   |   |-- task-status-selector.scss
|   |   |   \-- task-status-selector.tpl.html
|   |   |-- task-submission-viewer
|   |   |   |-- task-submission-viewer.coffee
|   |   |   |-- task-submission-viewer.scss
|   |   |   \-- task-submission-viewer.tpl.html
|   |   |-- task-submission-wizard
|   |   |   |-- task-submission-wizard.coffee
|   |   |   |-- task-submission-wizard.scss
|   |   |   \-- task-submission-wizard.tpl.html
|   |   |-- task-viewer
|   |   |   |-- task-viewer.coffee
|   |   |   |-- task-viewer.scss
|   |   |   \-- task-viewer.tpl.html
|   |   \-- tasks.coffee
|   |-- units
|   |   |-- modals
|   |   |   |-- modals.coffee
|   |   |   |-- unit-create-modal
|   |   |   |   |-- unit-create-modal.coffee
|   |   |   |   \-- unit-create-modal.tpl.html
|   |   |   |-- unit-ilo-edit-modal
|   |   |   |   |-- unit-ilo-edit-modal.coffee
|   |   |   |   \-- unit-ilo-edit-modal.tpl.html
|   |   |   |-- unit-mark-submissions-offline-modal
|   |   |   |   |-- unit-mark-submissions-offline-modal.coffee
|   |   |   |   \-- unit-mark-submissions-offline-modal.tpl.html
|   |   |   |-- unit-student-enrolment-modal
```

```
|   |   |   |   |-- unit-student-enrolment-modal.coffee
|   |   |   |   \-- unit-student-enrolment-modal.tpl.html
|   |   |   \-- unit-tutorial-edit-modal
|   |   |       |-- unit-tutorial-edit-modal.coffee
|   |   |       \-- unit-tutorial-edit-modal.tpl.html
|   |   |-- states
|   |   |   |-- edit-unit-view
|   |   |   |   |-- edit-unit-view.coffee
|   |   |   |   \-- edit-unit-view.tpl.html
|   |   |   |-- states.coffee
|   |   |   |-- teacher-view
|   |   |   |   |-- teacher-view.coffee
|   |   |   |   \-- teacher-view.tpl.html
|   |   |   \-- units-admin-view
|   |   |       |-- units-admin-view.coffee
|   |   |       \-- units-admin-view.tpl.html
|   |   |-- stats
|   |   |   |-- stats.coffee
|   |   |   |-- unit-achievement-stats
|   |   |   |   |-- unit-achievement-stats.coffee
|   |   |   |   \-- unit-achievement-stats.tpl.html
|   |   |   \-- unit-target-grade-stats
|   |   |       |-- unit-target-grade-stats.coffee
|   |   |       \-- unit-target-grade-stats.tpl.html
|   |   |-- unit-analytics-viewer
|   |   |   |-- unit-analytics-viewer.coffee
|   |   |   \-- unit-analytics-viewer.tpl.html
|   |   |-- unit-details-editor
|   |   |   |-- unit-details-editor.coffee
|   |   |   \-- unit-details-editor.tpl.html
|   |   |-- unit-groupset-editor
|   |   |   |-- unit-groupset-editor.coffee
|   |   |   \-- unit-groupset-editor.tpl.html
```

```
|   |   |-- unit-ilo-editor
|   |   |   |-- unit-ilo-editor.coffee
|   |   |   \-- unit-ilo-editor.tpl.html
|   |   |-- unit-staff-editor
|   |   |   |-- unit-staff-editor.coffee
|   |   |   \-- unit-staff-editor.tpl.html
|   |   |-- unit-student-list
|   |   |   |-- unit-student-list.coffee
|   |   |   |-- unit-student-list.scss
|   |   |   \-- unit-student-list.tpl.html
|   |   |-- unit-student-plagiarism-list
|   |   |   |-- unit-student-plagiarism-list.coffee
|   |   |   \-- unit-student-plagiarism-list.tpl.html
|   |   |-- unit-student-portfolio-list
|   |   |   |-- unit-student-portfolio-list.coffee
|   |   |   |-- unit-student-portfolio-list.scss
|   |   |   \-- unit-student-portfolio-list.tpl.html
|   |   |-- unit-students-editor
|   |   |   |-- unit-students-editor.coffee
|   |   |   \-- unit-students-editor.tpl.html
|   |   |-- unit-tasks-editor
|   |   |   |-- unit-tasks-editor.coffee
|   |   |   \-- unit-tasks-editor.tpl.html
|   |   |-- unit-tutorials-editor
|   |   |   |-- unit-tutorials-editor.coffee
|   |   |   \-- unit-tutorials-editor.tpl.html
|   |   \-- units.coffee
|   |-- users
|   |   |-- modals
|   |   |   |-- modals.coffee
|   |   |   |-- user-notification-settings-modal
|   |   |   |   |-- user-notification-settings-modal.coffee
|   |   |   |   \-- user-notification-settings-modal.tpl.html
```

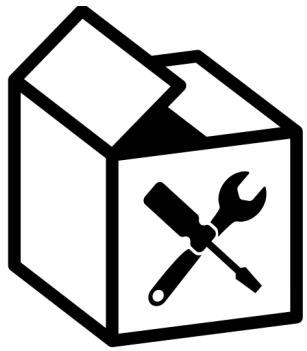
```
|   |   |   \-- user-settings-modal
|   |   |       |-- user-settings-modal.coffee
|   |   |       |-- user-settings-modal.tpl.html
|   |   |-- states
|   |   |   |-- states.coffee
|   |   |   \-- users-admin-view
|   |   |       |-- users-admin-view.coffee
|   |   |       |-- users-admin-view.tpl.html
|   |   \-- users.coffee
|   \-- visualisations
|       |-- achievement-box-plot.coffee
|       |-- achievement-custom-bar-chart.coffee
|       |-- alignment-bar-chart.coffee
|       |-- alignment-bullet-chart.coffee
|       |-- progress-burndown-chart.coffee
|       |-- progress-burndown-chart.scss
|       |-- student-task-status-pie-chart.coffee
|       |-- summary-task-status-scatter.coffee
|       |-- target-grade-pie-chart.coffee
|       |-- task-completion-box-plot.coffee
|       |-- task-status-pie-chart.coffee
|       |-- visualisation.tpl.html
|       \-- visualisations.coffee
|-- assets
|   |-- fonts
|   |   |-- bootstrap -> ../../../../vendor/bootstrap-sass/assets/fonts/bootstrap/
|   |   |-- font-awesome -> ../../../../vendor/font-awesome/fonts
|   |   \-- roboto
|   |       |-- Roboto-Bold.woff
|   |       |-- Roboto-Italic.woff
|   |       |-- Roboto-Light.woff
|   |       |-- Roboto-Medium.woff
|   |       \-- Roboto-Regular.woff
```

```
| -- images
|   |-- android-chrome-144x144.png
|   |-- android-chrome-192x192.png
|   |-- android-chrome-36x36.png
|   |-- android-chrome-48x48.png
|   |-- android-chrome-72x72.png
|   |-- android-chrome-96x96.png
|   |-- apple-touch-icon-114x114.png
|   |-- apple-touch-icon-120x120.png
|   |-- apple-touch-icon-144x144.png
|   |-- apple-touch-icon-152x152.png
|   |-- apple-touch-icon-180x180.png
|   |-- apple-touch-icon-57x57.png
|   |-- apple-touch-icon-60x60.png
|   |-- apple-touch-icon-72x72.png
|   |-- apple-touch-icon-76x76.png
|   |-- apple-touch-icon-precomposed.png
|   |-- apple-touch-icon.png
|   |-- doubtfire-logo-person-black.svg
|   |-- doubtfire-logo-person-white.svg
|   |-- favicon-16x16.png
|   |-- favicon-32x32.png
|   |-- favicon-96x96.png
|   |-- favicon.ico
|   |-- logo.svg
|   |-- manifest.json
|   |-- mask-icon.svg
|   |-- mrsdoubtfire.png
|   \-- person-unknown.gif
|-- common
|   |-- i18n
|   |   \-- localize.js
|   \-- utilService
```

```
|      \-- utilService.coffee
|-- i18n
|   |-- resources-locale_default.js
|   |-- resources-locale_en-AU.js
|   |-- resources-locale_en-GB.js
|   \-- resources-locale_en-US.js
|-- index.html
\-- styles
    |-- common
        |-- alerts.scss
        |-- center-full-screen.scss
        |-- extensions
            |-- container-fixed-x.scss
            |-- fa-document-o.scss
            |-- panel-footer-toolbar.scss
            |-- panel-heading-toolbar.scss
            |-- pointer.scss
            \-- strong.scss
        |-- grade-colors.scss
        |-- overrides
            |-- badge-overrides.scss
            |-- body-overrides.scss
            |-- header-overrides.scss
            |-- label-overrides.scss
            |-- nav-tabs-overrides.scss
            |-- nvd3-graph-overrides.scss
            |-- panel-overrides.scss
            |-- rating-overrides.scss
            \-- table-overrides.scss
        |-- task-status-colors.scss
        |-- typeface.scss
        \-- variables.scss
\-- config
```

```
|   \-- font-awesome.scss
|-- main.tmp.scss
|-- mixins
|   |-- animations
|   |   |-- fade-in.scss
|   |   |-- grow.scss
|   |   |-- slide-down.scss
|   |   \-- wobble.scss
|   |-- callout.scss
|   |-- dropdown-selector.scss
|   |-- flex-center.scss
|   |-- large-notice-block.scss
|   |-- logo-font-rendering.scss
|   |-- remove-list-padding.scss
|   \-- task-status-colors-generator.scss
|-- modules
|   |-- breadcrumbs.scss
|   |-- callout.scss
|   |-- doubtfire-logo.scss
|   |-- drilldown-visualiser.scss
|   |-- file-drop-zone.scss
|   |-- panel-fullscreen.scss
|   |-- project-task-bar.scss
|   |-- rationale-wrapper.scss
|   |-- tabset-icon.scss
|   \-- task-status.scss
\-- vendor-config
    |-- bootstrap.scss
    \-- font-awesome.scss
```

153 directories, 407 files



Chapter 12

Source Code

Contents

1 Doubtfire API Commits	2
2 Doubtfire Web Commits	3
3 Doubtfire API Source Code Changes	4
4 Doubtfire Web Source Code Changes	5

1 Doubtfire API Commits

See attached document

Commits on Jul 05, 2016

- o alexcu NEW: Add helpdesk ticket migration a5ab51d
- o alexcu NEW: Add basic Helpdesk Ticket model e89c95e

Commits on Jul 08, 2016

- o alexcu Merge branch 'config/update-dependencies' into new/helpdesk-ticketing f638b78
- o alexcu QUALITY: Change ticket comment to description ... 19fa45d
- o alexcu NEW: Add helpdesk ticket migration ed253dc
- o alexcu NEW: Add basic Helpdesk Ticket model 7d05ac3
- o alexcu QUALITY: Change ticket comment to description ... aa5ba09
- o alexcu ENHANCE: Add create_ticket permission to student projects 78d6a55
- o alexcu ENHANCE: Add convenience methods to helpdesk ticket model 5ecd064
- o alexcu NEW: Add POST /helpdesk/ticket e0e6195
- o alexcu QUALITY: Modularise helpdesk API into own module 9a00766
- o alexcu CONFIG: Mount helpdesk ticket API ebffaaa
- o alexcu Merge branch 'test/add-api-units-test' into new/helpdesk-ticketing ... e3a3c18
- o alexcu FIX: Ensure provided task matches project provided af73595

Commits on Jul 11, 2016

- o alexcu Merge branch 'develop' into new/helpdesk-ticketing e4a6449

Commits on Jul 12, 2016

- o alexcu CONFIG: Update schema.rb effe6c3
- o alexcu QUALITY: Move require_all helpers to relevant helpers area 5b6bcc8
- o alexcu TEST: Add HelpdeskTicket model test 8d4663a
- o jakerenzella Merge branch 'new/helpdesk-ticketing' of https://github.com/final-yea... ... 17d6f6a
- o jakerenzella TEST: Rename ticket.rb to ticket_test.rb ... 0e95a43
- o jakerenzella Fix: Fix asserts to correctly use assert, assert_equal, refute ... ce22e0d

Commits on Jul 21, 2016

- o alexcu Merge branch 'develop' into new/helpdesk-ticketing fee8936
- o jakerenzella FIX: Invalid create syntax for HelpdeskTicket test 5373312
- o jakerenzella NEW: Add GET /helpdesk/ticket a7c8fc2
- o jakerenzella FIX: Add missing permissions to helpdesk ticket 68b7aac
- o jakerenzella CONFIG: Update schema.rb 9b2e195
- o jakerenzella Merge branch 'develop' into new/helpdesk-ticketing be3db7a
- o alexcu NEW: Create ticket generator 8d7c19f
- o jakerenzella Merge remote-tracking branch 'upstream/develop' into new/helpdesk-tic... ... f5e0674
- o alexcu NEW: Add generator for helpdesk tickets to populator f44c3f2
- o jakerenzella QUALITY: Remove unnecessary dependencies 43c0041
- o jakerenzella QUALITY: Remove unnecessary dependencies ff34e40
- o jakerenzella FEATURE: Implement helpdesk_ticket endpoints 817e10b
- o jakerenzella QUALITY: Add role type users 0e6a56b
- o jakerenzella TEST: Add helpdesk_ticket endpoints tests 1a3a323
- o jakerenzella Merge branch 'new/helpdesk-ticketing' of https://github.com/final-yea... ... f88ebc2
- o jakerenzella QUALITY: Change auth_token to use isntance variable 24f7e73

Commits on Jul 22, 2016

- o alexcu NEW: Add generator for helpdesk tickets to populator bb9dde7
- o alexcu QUALITY: Remove debugging message from test 9456c4f
- o alexcu QUALITY: Move find_or_create_student into own helper 83b3292
- o alexcu NEW: Add randomizer class helper for sampling models b79902d

alexcu	FIX: Ensure ticket tests use random projects and tasks	5694ef6
alexcu	TEST: Remove unnessasary resolution tests	abbfb08
alexcu	QUALITY: Add auth token helper methods to tests	0a10dcf
alexcu	FIX: Add missing module function for with_auth_token	83a4ab4
jakerenzella	QUALITY: Make endpoint tests use auth helpers	2f25cae
jakerenzella	Merge branch 'new/helpdesk-ticketing' of https://github.com/final-yea... ...	e07103b
jakerenzella	FIX: Remove bad auth method calls	342bd9e
alexcu	QUALITY: Remove debug messages	841a934
alexcu	QUALITY: Add extra json helpers ...	4e9c32b
alexcu	FIX: Amend ticket API to support improved filtering ...	20367e8
alexcu	NEW: Add helpdesk sessions migration	5bbc1ae
alexcu	NEW: Add Helpdesk Session model	4c81c72
alexcu	ENHANCE: Add convenience methods for active sessions/users	01d83a6
alexcu	NEW: Add helpdesk session API	ddfa0a8

Commits on Jul 23, 2016

jakerenzella	QUALITY: Remove unnecessary test_helper code	6b39001
--------------	--	---------

Commits on Jul 25, 2016

jakerenzella	QUALITY: Consolidate auth_token_for_user into auth_token	7d6bbcf
jakerenzella	FIX: Ensure add_auth_token respects user provided	7711d24
jakerenzella	ENHANCE: Add randomizer for task definition	6acc8d9
jakerenzella	FIX: Replace t with ticket in ticket api logger	3f54d06
jakerenzella	QUALITY: Use create! instead of do	9dcda7a
jakerenzella	QUALITY: Allow Project.task_for_task_definition to accept an ID	f392258
jakerenzella	FIX: Remove floating find from ticket api	de25f58
jakerenzella	TEST: Write post test for new ticket	1409950

Commits on Jul 26, 2016

alexcu	NEW: Add resolved_at time to helpdesk ticket schema	fcc925e
alexcu	QUALITY: Add resolution time to ticket generator	4d95b69
alexcu	NEW: Add minutes to resolve to ticket schema	e3553c7
alexcu	QUALITY: Add minutes to resolve to database ticket populator	9d5a019
alexcu	NEW: Add average minutes to resolve ticket between time	5722487

Commits on Aug 07, 2016

alexcu	ENHANCE: Add tickets resolved between dates	03a4c60
alexcu	NEW: Add GET /helpdesk/stats	d9d7cab
alexcu	ENHANCE: Enforce session modification for session user only ...	2220555
alexcu	QUALITY: Update API to support changed permissions	b40a400
alexcu	QUALITY: Pluralise helpdesk session endpoints	5626d75
alexcu	NEW: Add GET /helpdesk/sessions	2d38a91
alexcu	ENHANCE: Add improved session serializer	54f3ab8
alexcu	QUALITY: Pluralize helpdesk ticket endpoints	08c68f5
alexcu	QUALITY: Bump number of generated helpdesk tickets	aed50a7
alexcu	ENHANCE: Update helpdesk ticket serializer	d01e41f
alexcu	Merge branch 'new/helpdesk-sessions' into new/helpdesk-ticketing	4571199
alexcu	QUALITY: Add TODOs for improved session authentication	dc62200
alexcu	QUALITY: Move stats endpoint into own API module	1b98cdf
alexcu	ENHANCE: Add units taught to all staff on helpdesk endpoint	996b36a
alexcu	NEW: Add average session times and counts to stats API	6462053

Commits on Aug 16, 2016

-o  Reubsinit	NEW: Add tickets for user API endpoint	3c39e39
 Commits on Aug 22, 2016		
-o  lachlanwest	ENHANCE: Combines 2 ticket API endpoints ...	91f3761
 Commits on Aug 23, 2016		
-o  Reubsinit	FIX: Replace task_id with task_definition_id	5fce6b8
 Commits on Aug 24, 2016		
-o  alexcu	Merge pull request #1 from final-year-project/api/add-user-ticket-end... ...	c88982d
 Commits on Aug 27, 2016		
-o  alexcu	QUALITY: Remove debugging messages	73510c4
-o  alexcu	FIX: Prevent creating a new ticket when user has one open	7251ad8
-o  alexcu	NEW: Add closed state to helpdesk ticket	7c5e39e
-o  alexcu	TEST: Fix failing Helpdesk tests	70a33f2
-o  alexcu	TEST: Add tests for DELETE ticket endpoint	75716ab
-o  alexcu	QUALITY: Add missing is_closed field to ticket serialiser	d5ad1d1
 Commits on Aug 29, 2016		
-o  alexcu	FIX: Fix error attempting to multiply nil	c881e5d
-o  alexcu	FIX: Add missing project ID to ticket serializer	e8b5f09
-o  alexcu	FIX: Fix mispelt method name resolved_between	66fe52d
-o  alexcu	FIX: Amend incorrect selection logic for resolved tickets	f513e8b
-o  alexcu	FIX: Stats for sessions return user_id only	09bf98b
-o  alexcu	FIX: Allow students to get tickets of their own	4c7fa84
-o  alexcu	FIX: Add missing task def id to ticket serialiser	37d0e25
 Commits on Sep 10, 2016		
-o  alexcu	QUALITY: Improve stats calculation	763012f
-o  alexcu	FIX: Add missing HelpdeskSessions class call	467485c
-o  alexcu	ENHANCE: Change the serializing of tickets	3be8a02
-o  alexcu	FIX: Ensure helpdesk user serialiser is used	a291cca
-o  alexcu	FIX: Ensure tickets are in order by creation time	53d4221
-o  alexcu	ENHANCE: Deprecate resolved_at for closed_at ...	eed08d1
 Commits on Sep 11, 2016		
-o  alexcu	ENHANCE: Revamp statistics endpoint	d8a83d6
-o  alexcu	FIX: Add new interval handling	6a90eed
 Commits on Sep 14, 2016		
-o  alexcu	ENHANCE: Add dashgraph endpoint to stats	4094807

2 Doubtfire Web Commits

See attached document

	Commits on Aug 03, 2016	
·	alexcu	L00KS: Remove gigantic buttons for selecting grade icon
		078493e
	Commits on Aug 12, 2016	
·	macite	FIX: Ensure Quality points > 0 to show details in task view
·	macite	FIX: Ensure task sheet can be viewed
·	macite	FIX: Typo in task status text
·	lachlanwest	NEW: Add help desk menu item to header menu
·	Reubsinit	NEW: Add directory structure for helpdesk ticket modal
·	Reubsinit	NEW: Add boiler plate code for helpdesk ticket modal
·	lachlanwest	NEW: Basic HTML structure of create ticket form
·	Reubsinit	NEW: Propagate prototype modal through help desk drop down
·	Reubsinit	NEW: Inject projects into scope of ticket modal
·	Reubsinit	NEW: Add all units to submit ticket unit dropdown
·	Reubsinit	NEW: Add task dropdown to ticket modal
		b10c7aa
	Commits on Aug 15, 2016	
·	Reubsinit	NEW: Add HelpDeskTicket to API models
·	Reubsinit	NEW: Modal now saves ticket to server
		c1ae833
		8e65b26
	Commits on Aug 23, 2016	
·	Reubsinit	NEW: Modal now saves task definition id to server
		cb4ae05
	Commits on Aug 27, 2016	
·	alexcu	Merge branch 'feature/helpdesk-ticket-modal' into feature/helpdesk-ti... ...
·	alexcu	QUALITY: Refactor helpdesk dropdown into directive
·	alexcu	NEW: Add additional API helpdesk endpoints
·	alexcu	QUALITY: Refactor submit modal
·	alexcu	FIX: Ensure query is used for custom state ticket requests
·	alexcu	QUALITY: Replace helpdesk ticket open bool for actual data
·	alexcu	ENHANCE: Show pill when ticket is open under header
·	alexcu	ENHANCE: Make submit ticket modal into generic ticket modal
·	alexcu	ENHANCE: Automatically add/remove ticket icon as needed
		896826a
		ece29e7
		dbc7dcc
		3102231
		6622d84
		26b6f2f
		239e574
		585635a
		fa031d9
	Commits on Aug 28, 2016	
·	alexcu	NEW: Add helpdesk session API endpoints
·	alexcu	FIX: Fix logic for loading task definition on ticket modal
·	alexcu	NEW: Add session modal
·	alexcu	FIX: Fix right global nav from squishing contents up
·	alexcu	ENHANCE: Place alerts underneath navbar
		617858d
		debd220
		2eb903c
		3da0ad9
		4f12b31
	Commits on Aug 29, 2016	
·	alexcu	FIX: Return from current session and ticket funcs on error
·	alexcu	NEW: Add HelpdeskStats factory
·	alexcu	QUALITY: Add find project helper method to project service
·	alexcu	NEW: Add partially implemented helpdesk-ticket directive
·	alexcu	NEW: Add partially implemented helpdesk dashboard state
·	alexcu	ENHANCE: Add link to helpdesk state in header dropdown
·	alexcu	QUALITY: Move stats polling into dashboard functionality
·	alexcu	FIX: Ensure ticket header data loads only if data is present
·	alexcu	QUALITY: Merge data models inside ticket-modal to ngResource
·	alexcu	FIX: Add missing ngModel attribute to task-def-selector
		b12f363
		e9fcebc
		99d7dc6
		1599cc6
		54a923c
		c52d801
		65fb6c9
		717d447
		6dfba40
		4f64fcf

-o	 alexcu	FIX: Ensure polling stops when dashboard controller dies	80341c5
 Commits on Sep 10, 2016			
-o	 alexcu	FIX: Stop flash issue on dashboard for tickets	1bab857
-o	 alexcu	NEW: Implement wireframe prototype	fd4ada1
-o	 alexcu	ENHANCE: Add position number to enqueued tickets	9dace26
-o	 alexcu	FIX: Fix incorrect logic for color warning ranges	78158a3
-o	 alexcu	FIX: Ensure time to resolve is rounded	0cc3237
-o	 alexcu	QUALITY: Move ticket position to dashboard only	63acb68
-o	 alexcu	LOOKS: Centre align tickets on dash	f0ee9b8
-o	 alexcu	FIX: Fix breaking create ticket without task	bb72dc3
 Commits on Sep 11, 2016			
-o	 alexcu	NEW: Add graph	400a58d
 Commits on Sep 14, 2016			
-o	 alexcu	FIX: Switch to dashgraph endpoint for graph	b154a6a
 Commits on Sep 18, 2016			
-o	 alexcu	FIX: Fix gruntfile leading comma	cb83dcc
-o	 alexcu	LOOKS: Improve headerbar for mobile devices	9af47f2
-o	 alexcu	LOOKS: Improve ticket display for mobile devices	a4cd61
-o	 alexcu	LOOKS: Show spinner during load of dashboard	4df0412
-o	 alexcu	LOOKS: Make buttons in modals clickable on mobile	049cd0a
-o	 alexcu	LOOKS: Improve looks for the dashboard menu	42d6b13
-o	 alexcu	LOOKS: Hide x overflow for tickets only	e1458dc
-o	 alexcu	FIX: Poll on clock on of helpdesk if done via dashboard	ba758d7
-o	 alexcu	FIX: Remove overflow on ticket row	eda760c
 Commits on Sep 19, 2016			
-o	 lachlanwest	ENHANCE: Make the dash emoji depend on wait time	1bb198a
 Commits on Sep 20, 2016			
-o	 alexcu	Merge pull request #1 from final-year-project/enhance/status-dependen... ...	a004fd4
-o	 alexcu	Merge pull request #2 from final-year-project/test/usability ...	947039b

3 Doubtfire API Source Code Changes

See attached document

4 Gemfile

View

```

@@ -44,12 +44,12 @@ group :production, :test, :replica do
 44   44   end
 45   45
 46   46   group :development, :test, :replica do
 47 -  gem 'rspec-rails', '~> 3'
 48 +  gem 'factory_girl_rails'
 49   48   gem 'minitest-rails'
 50   49   gem 'minitest-hyper'
 51   50   gem 'database_cleaner'
 52 -  gem "minitest-osx"
 51 +  gem 'minitest-osx'
 52 +  gem 'minitest-around'
 53   53   end
 54   54
 55   55   # Student submission

```

22 Gemfile.lock

View

```

@@ -81,7 +81,6 @@ GEM
 81   81     devise_ldap_authenticatable (0.8.5)
 82   82       devise (>= 3.4.1)
 83   83       net-ldap (>= 0.6.0, <= 0.11)
 84 -  diff-lcs (1.2.5)
 85   84       docile (1.1.5)
 86   85       encryptor (1.3.0)
 87   86       enumerable-lazy (0.0.1)
@@ -129,6 +128,8 @@ GEM
 129  128       mime-types-data (3.2016.0521)
 130  129       mini_portile2 (2.1.0)
 131  130       minitest (5.9.0)
 131 +  minitest-around (0.3.2)
 132   132       minitest (~> 5.0)
 132  133       minitest-hyper (0.2.0)
 133  134       minitest-osx (0.1.0)
 134   135       minitest (~> 5.4)
@@ -199,23 +200,6 @@ GEM
 199  200       responders (2.2.0)
 200  201       railties (>= 4.2.0, < 5.1)
 201  202       rmagick (2.15.4)
 202 -  rspec-core (3.4.4)
 203 -  rspec-support (~> 3.4.0)
 204 -  rspec-expectations (3.4.0)
 205 -  diff-lcs (>= 1.2.0, < 2.0)
 206 -  rspec-support (~> 3.4.0)
 207 -  rspec-mocks (3.4.1)
 208 -  diff-lcs (>= 1.2.0, < 2.0)
 209 -  rspec-support (~> 3.4.0)
 210 -  rspec-rails (3.4.2)
 211 -  actionpack (>= 3.0, < 4.3)
 212 -  activesupport (>= 3.0, < 4.3)
 213 -  railties (>= 3.0, < 4.3)
 214 -  rspec-core (~> 3.4.0)
 215 -  rspec-expectations (~> 3.4.0)
 216 -  rspec-mocks (~> 3.4.0)
 217 -  rspec-support (~> 3.4.0)
 218 -  rspec-support (3.4.1)
 219  203       ruby-filemagic (0.7.1)
 220  204       ruby-progressbar (1.8.1)

```

```

221  205      rubyzip (1.2.0)
  @@ -273,6 +257,7 @@ DEPENDENCIES
273  257      grape-swagger
274  258      hirb
275  259      launchy
+  260      minitest-around
276  261      minitest-hyper
277  262      minitest-osx
278  263      minitest-rails
  @@ -287,7 +272,6 @@ DEPENDENCIES
287  272      rails_best_practices
288  273      require_all (= 1.3.3)
289  274      rmagick (~> 2.15)
290  -  275      rspec-rails (~> 3)
291  275      ruby-filemagic
292  276      rubyzip
293  277      simplecov
  @@

```

6 app/api/api.rb

View
Monitor

```

  @@ -32,6 +32,9 @@ class Root < Grape::API
32   32      mount Api::Submission::PortfolioApi
33   33      mount Api::Submission::PortfolioEvidenceApi
34   34      mount Api::Submission::BatchTask
35 +  35      mount Api::Helpdesk::Ticket
36 +  36      mount Api::Helpdesk::Session
37 +  37      mount Api::Helpdesk::Stats
35   38
36   39      #
37   40      # Add auth details to all end points
  @@ -51,6 +54,9 @@ class Root < Grape::API
51   54      AuthHelpers.add_auth_to Api::Submission::PortfolioApi
52   55      AuthHelpers.add_auth_to Api::Submission::PortfolioEvidenceApi
53   56      AuthHelpers.add_auth_to Api::Submission::BatchTask
57 +  57      AuthHelpers.add_auth_to Api::Helpdesk::Ticket
58 +  58      AuthHelpers.add_auth_to Api::Helpdesk::Session
59 +  59      AuthHelpers.add_auth_to Api::Helpdesk::Stats
54   60
55   61      add_swagger_documentation \
56   62          base_path: nil,
  @@

```

104 app/api/helpdesk/session.rb

View
Monitor

```

...
...
@@ -0,0 +1,104 @@
1 +require 'grape'
2 +
3 +module Api
4 +  module Helpdesk
5 +    #
6 +    # Helpdesk session system endpoints
7 +    #
8 +    class Session < Grape::API
9 +      helpers AuthHelpers
10 +     helpers AuthorisationHelpers
11 +     helpers LogHelper
12 +
13 +     before do
14 +       authenticated?
15 +     end
16 +

```

```
17 + # -----
18 + # GET /helpdesk/sessions?user_id=[id]&is_active=[true|false]
19 + #
20 + desc "Get helpdesk sessions"
21 + params do
22 +   optional :user_id, type: Integer, :desc => "Filter by specific user id"
23 +   optional :is_active, type: Boolean, :desc => "Filter to only active sessions"
24 + end
25 + get '/helpdesk/sessions' do
26 +   unless authorise? current_user, HelpdeskSession, :get_sessions
27 +     error!({"error" => "Not authorised get helpdesk sessions"}, 403)
28 +   end
29 +   if params.empty?
30 +     logger.info "#{current_user.username} requested all helpdesk sessions"
31 +     HelpdeskSession.all
32 +   else
33 +     user_id = params[:user_id]
34 +     is_active = params[:is_active]
35 +     logger.info "#{current_user.username} requested all #{is_active ? 'active' : ''} helpdesk sessions" << (!user_
36 +     sessions = is_active ? HelpdeskSession.active_sessions : HelpdeskSession.all
37 +     sessions = user_id.nil? ? sessions : sessions.where(user_id: user_id)
38 +     sessions
39 +   end
40 + end
41 +
42 + #
43 + # POST /helpdesk/sessions
44 + #
45 + desc "Begin a new session at the helpdesk as current user"
46 + params do
47 +   requires :clock_off_time, type: DateTime, :desc => "The estimated clock off time when the user ends working"
48 +   optional :user_id, type: Integer, :desc => "The user who is about to begin working (if blank, posting user is u_
49 + end
50 + post '/helpdesk/sessions' do
51 +   unless authorise? current_user, HelpdeskSession, :create_session
52 +     error!({"error" => "Not authorised to create a helpdesk session"}, 403)
53 +   end
54 +   user = params[:user_id] ? User.find(params[:user_id]) : current_user
55 +   # TODO: if current_user != user_id passed in then require PIN
56 +   if HelpdeskSession.user_clocked_off?(user)
57 +     session = HelpdeskSession.create!(
58 +       user: user,
59 +       clock_on_time: DateTime.now, # Clock on immediately when ticket is created
60 +       clock_off_time: params[:clock_off_time]
61 +     )
62 +     logger.info "#{current_user.username} created new helpdesk session (id=#{session.id})"
63 +   else
64 +     error!({"error" => "#{user.username} is already clocked on at the helpdesk"}, 403)
65 +   end
66 +   session
67 + end
68 +
69 + #
70 + # DELETE /helpdesk/sessions/:id
71 + #
72 + desc "Prematurely clock off an existing helpdesk session"
73 + params do
74 +   requires :id, type: Integer, :desc => "The session to clock off"
75 + end
76 + delete '/helpdesk/sessions/:id' do
77 +   session = HelpdeskSession.find(params[:id])
78 +   unless authorise? current_user, session, :clock_off_session
79 +     error!({"error" => "Not authorised to clock off helpdesk session (id=#{session.id})"}, 403)
```

```

80      end
81      # TODO: if current_user != session.user then require PIN
82      if session.clocked_off?
83          logger.info "#{current_user.username} attempted to clock off already clocked off helpdesk session (id=#{session.id})"
84      else
85          session.clock_off
86          logger.info "#{current_user.username} prematurely clocked off helpdesk session (id=#{session.id})"
87      end
88      session
89  end
90
91  # -----
92  # GET /helpdesk/sessions/tutors
93  #
94  desc "Get a list of all currently tutors working at the helpdesk"
95  get '/helpdesk/sessions/tutors' do
96      unless authorise? current_user, HelpdeskSession, :get_all_current_session_users
97          error!({error: "Not authorised view current helpdesk staff"}, 403)
98      end
99      logger.info "#{current_user.username} requested all currently working helpdesk staff"
100     ActiveModel::ArraySerializer.new(HelpdeskSession.users_working_now, each_serializer: HelpdeskUserSerializer)
101 end
102 end
103 end
104 end

```

76 app/api/helpdesk/stats.rb

View
Code

```

...
...
@@ -0,0 +1,76 @@
1  +require 'grape'
2
3  +module Api
4  +  module Helpdesk
5  +    #
6  +    # Helpdesk ticket stats endpoints
7  +    #
8  +    class Stats < Grape::API
9  +        helpers AuthHelpers
10 +       helpers AuthorisationHelpers
11 +       helpers LogHelper
12 +
13 +       before do
14 +           authenticated?
15 +       end
16 +
17 +       # -----
18 +       # GET /helpdesk/stats/tickets
19 +       #
20 +       desc 'Gets tickets stats for helpdesk within the duration specified'
21 +       params do
22 +           optional :from, type: DateTime, desc: 'The time to start getting statistics (do not provide for all stats)'
23 +           optional :to,   type: DateTime, desc: 'The time to stop getting statistics (defaults to current time)', default: DateTime.now
24 +       end
25 +       get '/helpdesk/stats/tickets' do
26 +           unless authorise? current_user, HelpdeskTicket, :get_ticket_stats
27 +               error!({error: 'Not authorised to get helpdesk ticket stats'}, 403)
28 +           end
29 +
30 +           logger.info "#{current_user.username} got ticket helpdesk statistics"
31 +
32 +           from    = params[:from]
33 +           to     = params[:to] || DateTime.now
34 +

```

```

35 +     {
36 +         resolved_count: HelpdeskTicket.resolved_between(from, to).length,
37 +         number_unresolved: HelpdeskTicket.all_unresolved.length,
38 +         average_wait_time_in_mins: HelpdeskTicket.average_wait_time(from, to)
39 +     }
40 +   end
41 +
42 + # -----
43 + # GET /helpdesk/stats/dashgraph
44 + #
45 + desc 'Gets dashboard graph data for the helpdesk (last 3 hours)'
46 + get '/helpdesk/stats/dashgraph' do
47 +   unless authorise? current_user, HelpdeskTicket, :get_ticket_stats
48 +     error!({error: 'Not authorised to get helpdesk ticket stats'}, 403)
49 +   end
50 +
51 +   logger.info "#{current_user.username} got ticket helpdesk dash graph data"
52 +
53 +   # Work from now downto 3 hours ago
54 +   graph_range = 3.hours.ago.to_i..DateTime.now.to_i
55 +   graph_data = {}
56 +   # Collect data every minute going backwards -- stats for every min in
57 +   # last 180 mins
58 +   interval = 1.minute.to_i
59 +   graph_range.first.step(graph_range.last, interval) do |unix_t|
60 +     # Range is the time between graph_time less time mins ago
61 +     # to graph_time
62 +     range = Time.at(unix_t + interval).utc..Time.at(unix_t).utc
63 +     # Stats to include
64 +     avg_wait = HelpdeskTicket.average_wait_time(range.first, range.last)
65 +     unresolved = HelpdeskTicket.unresolved_between(range.first, range.last)
66 +     # Insert at this time the stats there were
67 +     graph_data[unix_t] = {
68 +       average_wait_time_in_mins: avg_wait,
69 +       number_of_unresolved_tickets: unresolved.length # count only
70 +     }
71 +   end
72 +   graph_data
73 + end
74 + end
75 + end
76 +end

```

120 app/api/helpdesk/ticket.rb

[View](#)

```

...
@@ -0,0 +1,120 @@
1 +require 'grape'
2 +require 'helpdesk_ticket_serializer'
3 +
4 +module Api
5+  module Helpdesk
6+    #
7+    # Helpdesk ticketing system endpoints
8+    #
9+    class Ticket < Grape::API
10+      helpers AuthHelpers
11+      helpers AuthorisationHelpers
12+      helpers LogHelper
13+
14+      before do
15+        authenticated?
16+      end
17+

```

```

18 + # -----
19 + # POST /helpdesk/tickets
20 +
21 + # -----
22 + desc "Add a new helpdesk ticket"
23 + params do
24 +   requires :project_id, type: Integer, desc: "The project to assign the ticket to"
25 +   optional :task_definition_id, type: Integer, desc: "Task which the student needs help with"
26 +   optional :description, type: String, desc: "Description associated to the ticket"
27 + end
28 + post '/helpdesk/tickets' do
29 +   project = Project.find(params[:project_id])
30 +
31 +   task = params[:task_definition_id].nil? ? nil : task = project.task_for_task_definition(params[:task_definition_id])
32 +
33 +   unless authorise? current_user, project, :create_ticket
34 +     error!({error: "Not authorised to create a ticket for project #{project.id}"}, 403)
35 +   end
36 +
37 +   # Only allow them to create a new ticket if they haven't done so already
38 +   if HelpdeskTicket.user_has_ticket_open? project.user.id
39 +     logger.info "#{current_user.username} tried to create new ticket but already has one open"
40 +     error!({error: "User already has a ticket open"}, 403)
41 +   end
42 +
43 +   ticket = HelpdeskTicket.create!(
44 +     project: project,
45 +     task: task,
46 +     description: params[:description]
47 +   )
48 +
49 +   logger.info "#{current_user.username} created new ticket #{ticket.id} for #{ticket.unit.name}"
50 +   ticket
51 +
52 + # -----
53 + # GET /helpdesk/tickets?filter={resolved,unresolved,closed,all}&user_id=id
54 + # Optional User Id and Filter. Filter default to all.
55 +
56 + desc "Gets all helpdesk tickets. Optional User Id and Resolved filter."
57 + params do
58 +   optional :user_id, type: String, desc: "The id of the user to get tickets for."
59 +   optional :filter, type: String, desc: "Filter by resolved, unresolved, closed or all. Defaults to all.", default: "all"
60 +   optional :shallow, type: Boolean, desc: "Use shallow serializer vs detailed serializer", default: true
61 + end
62 + get '/helpdesk/tickets' do
63 +   unless authorise? current_user, HelpdeskTicket, :get_tickets
64 +     error!({error: 'Not authorised to get tickets'}, 403)
65 +   end
66 +
67 +   user_id = params[:user_id]
68 +   filter = params[:filter] || 'all'
69 +   tickets = HelpdeskTicket.all_by_state(filter.to_sym, user_id)
70 +
71 +   serializer = params[:shallow] ? ShallowHelpdeskTicketSerializer : HelpdeskTicketSerializer
72 +   ActiveModel::ArraySerializer.new(tickets, each_serializer: serializer)
73 +
74 + # -----
75 + # GET /helpdesk/tickets/:id
76 +
77 + desc "Gets helpdesk ticket with an id"
78 + params do
79 +   requires :id, type: Integer, desc: "The id of the ticket to get"
80 + end

```

```

81 +     get '/helpdesk/tickets/:id' do
82 +       ticket = HelpdeskTicket.find(params[:id])
83 +
84 +       if not authorise? current_user, ticket, :get_details
85 +         error!({error: 'Not authorised to get ticket details'}, 403)
86 +       end
87 +
88 +       ticket
89 +     end
90 +
91 +   # -----
92 +   # DELETE /helpdesk/tickets/:id?[resolve=true|false]
93 +   #
94 +   desc "Updates helpdesk ticket with an id"
95 +   params do
96 +     requires :id, type: Integer, desc: "The id to of the ticket to delete"
97 +     optional :resolve, type: Boolean, desc: "Mark the ticket as resolved"
98 +   end
99 +   delete '/helpdesk/tickets/:id' do
100 +     ticket = HelpdeskTicket.find(params[:id])
101 +
102 +     if params[:resolve] == true
103 +       unless authorise? current_user, ticket, :resolve_ticket
104 +         error!({error: "Not authorised to resolve ticket #{params[:id]}"}, 403)
105 +       end
106 +       ticket.resolve
107 +       logger.info "#{current_user.username} resolved ticket #{ticket.id}"
108 +     else
109 +       unless authorise? current_user, ticket, :close_ticket
110 +         error!({error: "Not authorised to close ticket #{params[:id]}"}, 403)
111 +       end
112 +       ticket.close
113 +       logger.info "#{current_user.username} closed ticket #{ticket.id}"
114 +     end
115 +
116 +     ticket
117 +   end
118 + end
119 + end
120 +end

```

179 app/models/helpdesk_session.rb

[View](#)

```

...
...
@@ -0,0 +1,179 @@
1 +
2 ## Tracking for sessions of staff working at the helpdesk.
3 +
4 +class HelpdeskSession < ActiveRecord::Base
5 +  # Model associations
6 +  belongs_to :user
7 +
8 +  # Model constraints/validation
9 +  validates :user, presence: true
10 +  validates :clock_on_time, presence: true
11 +  validates :clock_off_time, presence: true
12 +
13 +  # Validates that a clock off time is always after the clock on time
14 +  validates_presence_of :clock_on_time, if: :clock_off_must_be_after_clock_on
15 +  def clock_off_must_be_after_clock_on
16 +    if clock_off_time && clock_off_time < clock_on_time
17 +      errors.add(:clock_off_time, "can't be before :clock_on_time")
18 +    end
19 +  end

```

```
20 +
21 + # Validates that the user for a session is not a student
22 + validates_presence_of :user, if: :user_is_not_student
23 + def user_is_not_student
24 +   if user && user.role == Role.student
25 +     errors.add(:user, 'must not be a student')
26 +   end
27 + end
28 +
29 + #
30 + # Permissions around helpdesk sessions
31 + #
32 + def self.permissions
33 +   # What can students do with sessions?
34 +   student_role_permissions = [
35 +     :get_all_current_session_users
36 +   ]
37 +   # What can tutors do with sessions?
38 +   tutor_role_permissions = [
39 +     :create_session,
40 +     :clock_off_session,
41 +     :get_all_current_session_users,
42 +     :get_sessions
43 +   ]
44 +   # What can convenors do with sessions?
45 +   convenor_role_permissions = [
46 +     :create_session,
47 +     :clock_off_session,
48 +     :get_all_current_session_users,
49 +     :get_sessions,
50 +     :get_ticket_stats
51 +   ]
52 +   # What can admins do with sessions?
53 +   admin_role_permissions = [
54 +     :create_session,
55 +     :clock_off_session,
56 +     :get_all_current_session_users,
57 +     :get_sessions,
58 +     :get_ticket_stats
59 +   ]
60 +   # What can nil users do with sessions?
61 +   nil_role_permissions = [
62 +   ]
63 +
64 +   # Return permissions hash
65 + {
66 +   :admin    => admin_role_permissions,
67 +   :convenor => convenor_role_permissions,
68 +   :tutor    => tutor_role_permissions,
69 +   :student  => student_role_permissions,
70 +   :nil      => nil_role_permissions
71 + }
72 + end
73 +
74 + def self.role_for(user)
75 +   user.role
76 + end
77 +
78 + def role_for(user)
79 +   # If the user provided is the user registered to the session, then
80 +   # they can do whatever they want with it. Convenors and admins can
81 +   # override this
82 +   if user == self.user || [Role.convenor, Role.admin].include?(user.role)
```

```
83 +     user.role
84 +   end
85 + end
86 +
87 + #
88 + # Override the clock off time to set it to now
89 + #
90 + def clock_off
91 +   self.clock_off_time = DateTime.now
92 +   save!
93 + end
94 +
95 + #
96 + # Returns the staff sessions within the given timeframe grouped by user id
97 + #
98 + def self.sessions_by_staff_between(from = nil, to = DateTime.now)
99 +   sessions_between(from, to).group_by(&:user)
100 + end
101 + def self.sessions_by_staff
102 +   sessions_by_staff_between
103 + end
104 +
105 + #
106 + # Returns all session stats between the timeframe
107 + #
108 + def self.stats_by_staff_id(from = nil, to = DateTime.now)
109 +   sessions_by_staff_between(from, to).map do |user, sessions|
110 +     durations = sessions.map(&:session_duration)
111 +     average_duration = durations.inject(:+) / durations.length
112 +     count = sessions.length
113 +     {
114 +       user_id: user.id,
115 +       average_duration_in_hours: average_duration,
116 +       count: count
117 +     }
118 +   end
119 + end
120 +
121 + #
122 + # Returns all sessions within the time-frame given
123 + #
124 + def self.sessions_between(from = nil, to = DateTime.now)
125 +   to ||= DateTime.now
126 +   return where('clock_on_time > ? AND clock_off_time < ?', from, to) unless from.nil?
127 +   where('clock_off_time < ?', to)
128 + end
129 +
130 + #
131 + # Returns all currently active sessions
132 + #
133 + def self.active_sessions
134 +   where('clock_off_time > ?', DateTime.now)
135 + end
136 +
137 + #
138 + # Returns all users currently working now
139 + #
140 + def self.users_working_now
141 +   active_sessions.map(&:user)
142 + end
143 +
144 + #
145 + # Returns true if this session is clocked off
```

```

146 + #
147 + def clocked_off?
148 +   clock_off_time < DateTime.now
149 + end
150 +
151 + #
152 + # Returns true if this session is clocked on
153 + #
154 + def clocked_on?
155 +   !clocked_off?
156 + end
157 +
158 + #
159 + # Checks if the user provided is currently clocked off
160 + #
161 + def self.user_clocked_off?(user)
162 +   # Should have no sessions clocked on thus nothing found clocked on
163 +   user.helpdesk_sessions.find(&:clocked_on?).nil?
164 + end
165 +
166 + #
167 + # Checks if the user provided is currently clocked on
168 + #
169 + def self.user_clocked_on?(user)
170 +   !user_clocked_off(user)
171 + end
172 +
173 + #
174 + # Returns the duration of time for a session in hours
175 + #
176 + def session_duration
177 +   (clock_off_time - clock_on_time) / 60 / 60
178 + end
179 +end

```

170 app/models/helpdesk_ticket.rb

[View](#)

```

...
@@ -0,0 +1,170 @@
+
## Tracking for tickets logged at the helpdesk.
#
+class HelpdeskTicket < ActiveRecord::Base
+  # Model associations
+  belongs_to :project
+  belongs_to :task
+
+  # Model constraints/validation
+  validates :project, presence: true # Must always be associated to a project
+
+  #
+  # Permissions around group data
+  #
+  def self.permissions
+    # What can students do with all tickets?
+    student_role_permissions = [
+      :get_details,
+      :get_tickets,
+      :get_ticket_stats,
+      :close_ticket
+    ]
+    # What can tutors|convenors|admins do with all tickets?
+    tutor_role_permissions = convenor_role_permissions = admin_role_permissions = [
+      :get_tickets,

```

```
26 +     :get_details,
27 +     :get_ticket_stats,
28 +     :close_ticket,
29 +     :resolve_ticket
30 +
31 +   # What can nil users do with all tickets?
32 +   nil_role_permissions = [
33 +
34 +   ]
35 +
36 +   # Return permissions hash
37 +   {
38 +     :admin    => admin_role_permissions,
39 +     :convenor => convenor_role_permissions,
40 +     :tutor    => tutor_role_permissions,
41 +     :student  => student_role_permissions,
42 +     :nil      => nil_role_permissions
43 +   }
44 + end
45 +
46 + def self.role_for(user)
47 +   user.role
48 + end
49 +
50 + def role_for(user)
51 +   if user == project.user
52 +     Role.student
53 +   elsif user.role != Role.student
54 +     user.role
55 +   else
56 +     nil
57 +   end
58 + end
59 +
60 + # Returns true if a user has a ticket open
61 + def self.user_has_ticket_open?(user_id)
62 +   !all_unresolved(user_id).empty?
63 + end
64 +
65 + # Returns back all unresolved tickets, optionally limit to a user
66 + def self.all_unresolved(user_id = nil)
67 +   all_by_state(:unresolved, user_id)
68 + end
69 +
70 + # Returns back all resolved tickets, optionally limit to a user
71 + def self.all_resolved(user_id = nil)
72 +   all_by_state(:resolved, user_id)
73 + end
74 +
75 + # Returns back all closed tickets, optionally limit to a user
76 + def self.all_closed(user_id = nil)
77 +   all_by_state(:closed, user_id)
78 + end
79 +
80 + # Finds tickets of a particular status, optionally limit to a user
81 + def self.all_by_state(resolved_filter, user_id = nil)
82 +   tickets =
83 +   case resolved_filter
84 +   when :resolved
85 +     where(is_resolved: true)
86 +   when :unresolved
87 +     where(is_closed: false)
88 +   when :closed
```

```
89      where(is_closed: true, is_resolved: false)
90    else
91      all
92    end
93  +
94  unless user_id.nil?
95    user = User.find(user_id)
96    project = Project.for_user(user, false) # see app/models/project.rb:76
97    tickets = tickets.where(project: project) # limits the scope of tickets down to those with the project provided
98  end
99 +
100 tickets.order(:created_at)
101 end
102 +
103 # Get all tickets resolved between two dates
104 def self.resolved_between(from = nil, to = DateTime.now)
105   to ||= DateTime.now # if nil is passed in
106   from ? all_resolved.where(closed_at: from..to) : all_resolved
107 end
108 +
109 #
110 # Get all tickets that were still unresolved between from and to
111 # dates
112 #
113 def self.unresolved_between(from, to = DateTime.now)
114   to ||= DateTime.now # if nil passed in
115   where('created_at <= ? AND (closed_at >= ? OR NOT is_closed)', from, to)
116 end
117 +
118 # Calculates the average time to resolve a ticket from the duration
119 # provided (i.e., between from and now). If no arguments are provided,
120 # all resolved tickets will be used regardless of when they were resolved.
121 # Where no tickets are found within the period, nil is returned.
122 def self.average_resolve_time_between(from = nil, to = DateTime.now)
123   tickets = resolved_between(from, to)
124   resolved_between(from, to).average(:minutes_to_resolve).to_i unless tickets.empty?
125 end
126 +
127 #
128 # Determines the current average wait time for a ticket
129 #
130 def self.average_wait_time(from, to)
131   avg_resolve = HelpdeskTicket.average_resolve_time_between(from, to) || 0
132   num_tickets = HelpdeskTicket.unresolved_between(from, to).length
133   avg_resolve * num_tickets
134 end
135 +
136 # Resolves the ticket
137 def resolve
138   unless is_closed
139     # Resolving closes the ticket
140     close
141     self.is_resolved = true
142     self.minutes_to_resolve = ((closed_at - created_at) / 60).to_f.round(2)
143     save!
144   end
145 end
146 +
147 # Unit for ticket
148 def unit
149   project.unit
150 end
151 +
```

```

152 + # Student for ticket
153 + def student
154 +   project.student
155 + end
156 +
157 + # Returns true if ticket is associated with a task
158 + def task?
159 +   !task.nil?
160 + end
161 +
162 + # Prematurely closes a ticket
163 + def close
164 +   unless is_closed
165 +     self.is_closed = true
166 +     self.closed_at = DateTime.now
167 +     save!
168 +   end
169 + end
170 +end

```

7 app/models/project.rb

View
Code

```

@@ -21,6 +21,8 @@ class Project < ActiveRecord::Base
21
22   has_many :learning_outcome_task_links, through: :tasks
23
24 + has_many :helpdesk_tickets, dependent: :destroy
25 +
26   validate :must_be_in_group_tutorials
27
28   #
@@ -33,7 +35,8 @@ def self.permissions
33   :changeTutorial,
34   :makeSubmission,
35   :getSubmission,
36   - :change
37   + :change,
38   + :createTicket
39 ]
40
# What can tutors do with projects?
41 tutor_role_permissions = [
@@ -860,6 +863,8 @@ def has_task_for_task_definition?(td)
860   # task if the task does not exist for this project.
861   #
862   def task_for_task_definition(td)
863     # Find the definition if a ID is passed in.
864     + td = TaskDefinition.find(td) if td.is_a? Fixnum
865     logger.debug "Finding task #{td.abbreviation} for project #{log_details()}"
866     result = tasks.where(task_definition: td).first
867     if result.nil?
868
869
870

```

1 app/models/task.rb

View
Code

```

@@ -74,6 +74,7 @@ def role_for(user)
74   has_many :reverse_plagiarism_match_links, class_name: "PlagiarismMatchLink", dependent: :destroy, inverse_of: :other_
75   has_many :learning_outcome_task_links, dependent: :destroy # links to learning outcomes
76   has_many :learning_outcomes, through: :learning_outcome_task_links
77 + has_many :helpdesk_tickets
78
79   validates :task_definition_id, uniqueness: { scope: :project,
80     message: "must be unique within the project" }

```

		View
5	app/models/user.rb	
	<pre> @@ -67,6 +67,7 @@ def reset_authentication_token! 67 belongs_to :role # Foreign Key 68 has_many :unit_roles, dependent: :destroy 69 has_many :projects 70 + has_many :helpdesk_sessions 71 # Model validations/constraints 72 validates :first_name, presence: true @@ -407,4 +408,8 @@ def self.import_from_csv(current_user, file) 408 errors: errors 409 } 410 end 411 + 412 + def units_taught 413 + unit_roles.map(&:unit) 414 + end 415 end </pre>	
15	app/serializers/helpdesk_session_serializer.rb	View
...	<pre> @@ -0,0 +1,15 @@ +require 'user_serializer' + +class HelpdeskSessionSerializer < ActiveModel::Serializer + attributes :id, + :user, + :clocked_on?, + :clock_on_time, + :clock_off_time + + has_one :user, serializer: HelpdeskUserSerializer + + def clocked_on? + object.clocked_on? + end +end </pre>	
40	app/serializers/helpdesk_ticket_serializer.rb	View
...	<pre> @@ -0,0 +1,40 @@ +require 'project_serializer' +require 'task_definition_serializer' + +class HelpdeskTicketSerializer < ActiveModel::Serializer + attributes :id, + :project, + :task_definition, + :description, + :is_resolved, + :is_closed, + :created_at, + :closed_at, + :minutes_to_resolve + + def project + ShallowProjectSerializer.new object.project + end + + def task_definition </pre>	

```

20 +   task = object.task
21 +   ShallowTaskDefinitionSerializer.new task.task_definition if task
22 + end
23 +
24 +
25 +class ShallowHelpdeskTicketSerializer < ActiveModel::Serializer
26 +  attributes :id,
27 +              :project_id,
28 +              :task_definition_id,
29 +              :description,
30 +              :is_resolved,
31 +              :is_closed,
32 +              :created_at,
33 +              :closed_at,
34 +              :minutes_to_resolve
35 +
36 +  def task_definition_id
37 +    task = object.task
38 +    task.task_definition_id if task
39 +  end
40 +end

```

25 app/serializers/project_serializer.rb

[View](#)

```

@@ -10,6 +10,31 @@ class ShallowProjectSerializer < ActiveModel::Serializer
10   def project_id
11     object.id
12   end
13 +
14 + def student_name
15 +   object.student.name
16 + end
17 +
18 + def unit_id
19 +   object.unit.id
20 + end
21 +
22 + def unit_code
23 +   object.unit.code
24 + end
25 +
26 + def unit_name
27 +   object.unit.name
28 + end
29 +
30 + def tutor_name
31 +   tutorial = object.tutorial
32 +   tutorial.tutor.name if tutorial
33 + end
34 +
35 + def start_date
36 +   object.unit.start_date.to_date
37 + end
38
39
40 # Student project serializer is used with teaching staff

```

4 app/serializers/task_definition_serializer.rb

[View](#)

```

...
...
@@ -1,3 +1,7 @@
1 +class ShallowTaskDefinitionSerializer < ActiveModel::Serializer
2 +  attributes :id, :abbreviation, :name, :description

```

	<pre> 3 +end 4 + 5 class TaskDefinitionSerializer < ActiveModel::Serializer 6 attributes :id, :abbreviation, :name, :description, 7 :weight, :target_grade, :target_date, 8 </pre>	
8	app/serializers/user_serializer.rb	View Code
	<pre> @@ -10,3 +10,11 @@ def system_role 10 class ShallowUserSerializer < ActiveModel::Serializer 11 attributes :id, :name, :email 12 end 13 + 14 +class HelpdeskUserSerializer < ActiveModel::Serializer 15 + attributes :id, :name, :email, :units_taught 16 + 17 + def units_taught 18 + ActiveModel::ArraySerializer.new(object.units_taught, each_serializer: ShallowUnitSerializer) 19 + end 20 +end </pre>	
15	db/migrate/20160705025331_create_helpdesk_tickets.rb	View Code
...	<pre> ... @@ -0,0 +1,15 @@ 1 +class CreateHelpdeskTickets < ActiveRecord::Migration 2 + def change 3 + create_table :helpdesk_tickets do t 4 + t.references :project, null: false 5 + t.references :task, null: true 6 + 7 + t.string :comments, null: true 8 + t.boolean :is_resolved, null: false, default: false 9 + 10 + t.timestamps 11 + end 12 + add_index :helpdesk_tickets, :project_id 13 + add_index :helpdesk_tickets, :task_id 14 + end 15 +end </pre>	
6	db/migrate/20160708011422_change_helpdesk_ticket_comment_to_description.rb	View Code
...	<pre> ... @@ -0,0 +1,6 @@ 1 +class ChangeHelpdeskTicketCommentToDescription < ActiveRecord::Migration 2 + def change 3 + rename_column :helpdesk_tickets, :comments, :description 4 + change_column :helpdesk_tickets, :description, :string, :limit => 2048 5 + end 6 +end </pre>	
11	db/migrate/20160722114921_create_helpdesk_sessions.rb	View Code
...	<pre> ... @@ -0,0 +1,11 @@ 1 +class CreateHelpdeskSessions < ActiveRecord::Migration 2 + def change 3 + create_table :helpdesk_sessions do t 4 + t.references :user, null: false 5 + t.datetime :clock_on_time, null: false 6 + t.datetime :clock_off_time, null: false 7 + t.timestamps 8 + end 9 + add_index :helpdesk_sessions, :user_id </pre>	

```
10 + end
11 +end
```

5 db/migrate/20160725141011_add_resolved_at_to_helpdesk_ticket.rb

[View](#) 

```
... ... @@ -0,0 +1,5 @@
1 +class AddResolvedAtToHelpdeskTicket < ActiveRecord::Migration
2 + def change
3 +   add_column :helpdesk_tickets, :resolved_at, :datetime
4 + end
5 +end
```

5 db/migrate/20160725144833_add_minutes_to_resolve_to_helpdesk_ticket.rb

[View](#) 

```
... ... @@ -0,0 +1,5 @@
1 +class AddMinutesToResolveToHelpdeskTicket < ActiveRecord::Migration
2 + def change
3 +   add_column :helpdesk_tickets, :minutes_to_resolve, :float
4 + end
5 +end
```

5 db/migrate/20160826233551_add_is_closed_to_helpdesk_ticket.rb

[View](#) 

```
... ... @@ -0,0 +1,5 @@
1 +class AddIsClosedToHelpdeskTicket < ActiveRecord::Migration
2 + def change
3 +   add_column :helpdesk_tickets, :is_closed, :boolean, :default => false
4 + end
5 +end
```

5 db/migrate/20160910065932_rename_resolved_at_to_closed_at.rb

[View](#) 

```
... ... @@ -0,0 +1,5 @@
1 +class RenameResolvedAtToClosedAt < ActiveRecord::Migration
2 + def change
3 +   rename_column :helpdesk_tickets, :resolved_at, :closed_at
4 + end
5 +end
```

27 db/schema.rb

[View](#) 

```
... @@ -11,7 +11,7 @@
11 #
12 # It's strongly recommended that you check this file into your version control system.
13
14 -ActiveRecord::Schema.define(version: 20160503222451) do
15 +ActiveRecord::Schema.define(version: 20160910065932) do
16
17   # These are extensions that must be enabled in order to support this database
18   enable_extension "plpgsql"
... @@ -74,6 +74,31 @@
74
75   add_index "helpdesk_schedules", ["user_id"], name: "index_helpdesk_schedules_on_user_id", using: :btree
76
77   + create_table "helpdesk_sessions", force: :cascade do |t|
78     + t.integer "user_id", null: false
79     + t.datetime "clock_on_time", null: false
80     + t.datetime "clock_off_time", null: false
81     + t.datetime "created_at"
82     + t.datetime "updated_at"
83   end
84
85   + add_index "helpdesk_sessions", ["user_id"], name: "index_helpdesk_sessions_on_user_id", using: :btree
```

```

86 +
87 +  create_table "helpdesk_tickets", force: :cascade do |t|
88 +    t.integer "project_id",                                     null: false
89 +    t.integer "task_id"
90 +    t.string  "description",        limit: 2048
91 +    t.boolean "is_resolved",       default: false, null: false
92 +    t.datetime "created_at"
93 +    t.datetime "updated_at"
94 +    t.datetime "closed_at"
95 +    t.float   "minutes_to_resolve"
96 +    t.boolean "is_closed",        default: false
97 +  end
98 +
99 +  add_index "helpdesk_tickets", ["project_id"], name: "index_helpdesk_tickets_on_project_id", using: :btree
100 + add_index "helpdesk_tickets", ["task_id"], name: "index_helpdesk_tickets_on_task_id", using: :btree
101 +
77 102   create_table "learning_outcome_task_links", force: :cascade do |t|
78    t.text    "description"
79    t.integer "rating"

```

View



131 lib/helpers/database_populator.rb

```

@@ -2,6 +2,7 @@
2   2   require 'faker'
3   3   require 'bcrypt'
4   4   require 'json'
5   5   +require_all 'lib/helpers'
6   6
7   7   #
8   8   # This class populates data in the database
@@ -29,7 +30,7 @@ def initialize(scale = :small)
29  30     some_tutorials: 1,
30  31     many_tutorials: 1,
31  32     max_tutorials: 4,
32  33   -   tickets_generated: 10
33  34   +   tickets_to_generate: 30
34  35   },
35  36   large: {
36  37     min_students: 15,
@@ -41,7 +42,7 @@ def initialize(scale = :small)
41  42     some_tutorials: 2,
42  43     many_tutorials: 4,
43  44     max_tutorials: 20,
44  45   -   tickets_generated: 50
45  46   +   tickets_to_generate: 100
46  47   }
47  48   accepted_scale_types = scale_data.keys
@@ -135,14 +136,80 @@ def generate_units
135 136     generate_and_align_ilos_for_unit(unit, unit_details)
136 137     generate_tutorials_and_enrol_students_for_unit(unit, unit_details)
137 138   end
138 139   +   puts "!"
139 140   end
140 141
141 142   #
142 143   - # Random project helper
143 144   + # Generates some units
144 145   #
145 146   - def random_project
146 147   -   id = Project.pluck(:id).sample

```

```

145      -   Project.find(id)
145 +  def generate_units
146 +  puts "--> Generating units"
147 +
148 +  if @user_cache.empty?
149 +    # Must generate users first!
150 +    puts "----> No users generated. Generating users first..."
151 +    generate_users()
152 +  end
153 +
154 +  # Set sizes from scale
155 +  some_tasks = @scale[:some_tasks]
156 +  many_tasks = @scale[:many_tasks]
157 +  some_tutorials = @scale[:some_tutorials]
158 +  many_tutorials = @scale[:many_tutorials]
159 +
160 +  # Run through the unit_details and initialise their data
161 +  @unit_data.each do | unit_key, unit_details |
162 +    puts "----> Generating unit #{unit_details[:code]}"
163 +    unit = Unit.create!(
164 +      code: unit_details[:code],
165 +      name: unit_details[:name],
166 +      description: Populator.words(10..15),
167 +      start_date: Time.zone.now - 6.weeks,
168 +      end_date: 13.weeks.since(Time.zone.now - 6.weeks)
169 +    )
170 +    # Assign the convenors for this unit
171 +    unit_details[:convenors].each do | user_key |
172 +      puts "----> Adding convenor #{user_key}"
173 +      unit.employ_staff(@user_cache[user_key], Role.convenor)
174 +    end
175 +    # Cache what we have
176 +    @unit_cache[unit_key] = unit
177 +    # Generate other unit-related stuff
178 +    generate_tasks_for_unit(unit, unit_details)
179 +    generate_and_align_ilos_for_unit(unit, unit_details)
180 +    generate_tutorials_and_enrol_students_for_unit(unit, unit_details)
181 +  end
182 +end
183 +
184 +#
185 +# Generates helpdesk tickets
186 +#
187 +def generate_helpdesk_tickets
188 +  tickets_to_generate = @scale[:tickets_to_generate]
189 +  print "> Generating #{tickets_to_generate} helpdesk tickets"
190 +  tickets_to_generate.times do
191 +    project = Randomizer.random_record_for_model(Project)
192 +    throw "Must create projects before calling generate_helpdesk_tickets" if project.nil?
193 +    # 1/4 chance of getting nil task
194 +    task = rand(0..3) == 0 ? nil : Randomizer.random_task_for_project(project)
195 +    # 3/4 chance of being resolved
196 +    is_resolved = rand(0..3) > 0
197 +    if is_resolved
198 +      minutes_to_resolve = rand(0..15)
199 +      closed_at = DateTime.now + minutes_to_resolve.minutes
200 +    end
201 +    HelpdeskTicket.create(
202 +      project: project,
203 +      task: task,
204 +      description: Populator.words(5..10),
205 +      is_resolved: is_resolved,
206 +      is_closed: is_resolved,

```

```

207     +     closed_at: closed_at,
208     +     minutes_to_resolve: minutes_to_resolve
209     +
210     +   print "."
211     + end
212     + puts "!"
146   213   end
147
214
148   215   private
216
217   @@ -323,42 +390,22 @@ def generate_tutorials_and_enrol_students_for_unit(unit, unit_details)
218
219   390   end
220
221   391
222   392   #
223
224   - # Finds or creates a student if not found
225   -
226   - #  

227   - def find_or_create_student(username)
228   -   unless @user_cache.has_key?(username)
229   -     profile = {
230   -       first_name: Faker::Name.first_name,
231   -       last_name: Faker::Name.last_name,
232   -       nickname: username,
233   -       role_id: Role.student_id,
234   -       email: "#{username}@doubtfire.com",
235   -       username: username,
236   -       password: 'password',
237   -       password_confirmation: 'password'
238   -     }
239   -     user = User.create!(profile)
240   -     @user_cache[username] = user
241   -   end
242   -   @user_cache[username]
243   - end
244
245
246
247   393   # Generated fixed data here for students and units
248   394   #
249   395   def generate_fixed_data
250   396     # Define fixed user data here
251   397     @user_data = {
252   398       acain: {first_name: "Andrew", last_name: "Cain", nickname: "Macite", role_id: Role},
253   399       cwoodward: {first_name: "Clinton", last_name: "Woodward", nickname: "The Giant", role_id: Role},
254   400       ajones: {first_name: "Allan", last_name: "Jones", nickname: "P-Jiddy", role_id: Role},
255   401       aconvenor: {first_name: "Clinton", last_name: "Woodward", nickname: "The Giant", role_id: Role},
256   402       aadmin: {first_name: "Allan", last_name: "Jones", nickname: "P-Jiddy", role_id: Role},
257   403       rwilson: {first_name: "Reuben", last_name: "Wilson", nickname: "Reubs", role_id: Role},
258   404       akihironoguchi: {first_name: "Akihiro", last_name: "Noguchi", nickname: "Animations", role_id: Role},
259   405       atutor: {first_name: "Akihiro", last_name: "Noguchi", nickname: "Animations", role_id: Role},
260   406       acummaudo: {first_name: "Alex", last_name: "Cummaudo", nickname: "DoubtfireDude", role_id: Role},
261   407       cliff: {first_name: "Cliff", last_name: "Warren", nickname: "Cliff", role_id: Role},
262   408       joostfunkekupper: {first_name: "Joost", last_name: "Funke Kupper", nickname: "Joe", role_id: Role},
263   409       angusmorton: {first_name: "Angus", last_name: "Morton", nickname: "Angus", role_id: Role},
264   410       "123456X" => {first_name: "Fred", last_name: "Jones", nickname: "Foo", role_id: Role},
265   411       "123456X" => {first_name: "Fred", last_name: "Jones", nickname: "Foo", role_id: Role},
266       astudent: {first_name: "student", last_name: "surname", nickname: "Foo", role_id: Role}
267     }
268
269     # Add 10 tutors to fixed info
270     10.times do |count|
271
272       @@ -381,14 +428,14 @@ def generate_fixed_data
273         intro_prog: {
274           code: "COS10001",
275           name: "Introduction to Programming",
276           convenors: [ :acain, :cwoodward ],
277           convenors: [ :acain, :aconvenor ],

```

```

385 432     tutors: [
386 433         { user: :acain, num: many_tutorials },
387 434         { user: :cwoodward, num: many_tutorials },
388 435         { user: :ajones, num: many_tutorials },
389 436         { user: :aconvenor, num: many_tutorials },
390 437         { user: :aadmin, num: many_tutorials },
391 438         { user: :rwilson, num: many_tutorials },
392 439         { user: :acummaudo, num: some_tutorials },
393 440         { user: :akihironoguchi, num: many_tutorials },
394 441         { user: :atutor, num: many_tutorials },
395 442         { user: :joostfunekupper, num: many_tutorials },
396 443         { user: :angusmorton, num: some_tutorials },
397 444         { user: :cliff, num: some_tutorials },
398 445
399 446     @@ -400,11 +447,11 @@ def generate_fixed_data
400 447         oop: {
401 448             code: "COS20007",
402 449             name: "Object Oriented Programming",
403 450             convenors: [ :acain, :cwoodward, :ajones, :acummaudo ],
404 451             +
405 452                 convenors: [ :acain, :aconvenor, :aadmin, :acummaudo ],
406 453                 tutors: [
407 454                     { user: "tutor_1", num: few_tutorials },
408 455                     { user: :angusmorton, num: few_tutorials },
409 456                     { user: :akihironoguchi, num: few_tutorials },
410 457                     { user: :atutor, num: few_tutorials },
411 458                     { user: :joostfunekupper, num: few_tutorials },
412 459                 ],
413 460                 num_tasks: many_tasks,
414 461
415 462     @@ -414,9 +461,9 @@ def generate_fixed_data
416 463         ai4g: {
417 464             code: "COS30046",
418 465             name: "Artificial Intelligence for Games",
419 466             convenors: [ :cwoodward ],
420 467             +
421 468                 convenors: [ :aconvenor ],
422 469                 tutors: [
423 470                     { user: :cwoodward, num: few_tutorials },
424 471                     { user: :aconvenor, num: few_tutorials },
425 472                     { user: :cliff, num: few_tutorials },
426 473                 ],
427 474                 num_tasks: few_tasks,
428 475
429 476     @@ -426,13 +473,13 @@ def generate_fixed_data
430 477         gameprog: {
431 478             code: "COS30243",
432 479             name: "Game Programming",
433 480             convenors: [ :cwoodward, :acummaudo ],
434 481             +
435 482                 convenors: [ :aconvenor, :acummaudo ],
436 483                 tutors: [
437 484                     { user: :cwoodward, num: few_tutorials },
438 485                     { user: :aconvenor, num: few_tutorials },
439                 ],
440                 num_tasks: few_tasks,
441                 ilos: rand(0..3),
442                 students: [ :acain, :ajones ]
443                 +
444                     students: [ :acain, :aadmin ]
445                 },
446             },
447         },
448     puts "-> Defined #{@user_data.length} fixed users and #{@unit_data.length} units"
449

```

22 lib/helpers/find_or_create_students.rb

[View](#)

***	***	@@ -0,0 +1,22 @@
	1	++#

```

2  ## Finds or creates a student if not found
3  ##
4  +def find_or_create_student(username)
5  + user_created = nil
6  + using_cache = !!@user_cache
7  + if not using_cache or not @user_cache.has_key?(username)
8  +   profile = {
9  +     first_name: Faker::Name.first_name,
10 +    last_name: Faker::Name.last_name,
11 +   nickname: username,
12 +   role_id: Role.student_id,
13 +   email: "#{username}@doubtfire.com",
14 +   username: username,
15 +   password: 'password',
16 +   password_confirmation: 'password'
17 + }
18 + user_created = User.create!(profile)
19 + @user_cache[username] = user_created if using_cache
20 + end
21 + user_created || @user_cache[username]
22 +end

```

25 lib/helpers/randomizer.rb

[View](#)

```

*** ...
1  @@ -0,0 +1,25 @@
2  +
3  +class Randomizer
4  + #
5  + # Randomly returns a new model (e.g., random_record_for_model(Project))
6  + #
7  + def self.random_record_for_model(model)
8  +   throw "Must provide a model" unless model.is_a? Class
9  +   id = model.all.pluck(:id).sample
10 +  model.find(id)
11 + end
12 +
13 + #
14 + # Randomly returns a new task for the given project
15 + #
16 + def self.random_task_for_project(project)
17 +   task_def = self.random_task_def_for_project(project)
18 +   project.task_for_task_definition(task_def)
19 + end
20 +
21 + #
22 + # Randomly returns a new task definition for the given project
23 + #
24 + def self.random_task_def_for_project(project)
25 +   project.unit.task_definitions.sample
26 + end
27 +end

```

26 lib/tasks/populate.rake

[View](#)

```

2  @@ -2,32 +2,11 @@ require_all 'lib/helpers'
3  namespace :db do
4    desc "Mark off some of the due tasks"
5    task expand_first_unit: :environment do
6      -
7      find_or_create_student = lambda { |username|
8        result = User.find_by_username(username)
9        if result
10          return result
11        else

```

```

11      -     profile = {
12          -         first_name:           Faker::Name.first_name,
13          -         last_name:            Faker::Name.last_name,
14          -         nickname:             username,
15          -         role_id:              Role.student_id,
16          -         email:                "#{username}@doubtfire.com",
17          -         username:              username,
18          -         password:              'password',
19          -         password_confirmation: 'password'
20      }
21
22      -     result = User.create!(profile)
23      -     return result
24  end
25 }
26
27 5     unit = Unit.first
28 6     tutes = unit.tutorials
29 7     for student_count in 0..2000
30     -     proj = unit.enrol_student(find_or_create_student.call("student_#{student_count}"), tutes[student_count % tutes.co
31 8     +     student = find_or_create_student("student_#{student_count}")
32 9     +     proj = unit.enrol_student(student, tutes[student_count % tutes.count])
33 10    end
34 11  end
35 12
36 @@ -177,6 +156,7 @@ namespace :db do
37 156    dbpop = DatabasePopulator.new ENV['SCALE']
38 157    dbpop.generate_users()
39 158    dbpop.generate_units()
40 159    + dbpop.generate_helpdesk_tickets()
41 160
42 161    # Run simulate signoff?
43 162    unless !args.nil? && args[:extend_populate]
44
45

```

2 lib/tasks/test_setup.rake

[View](#)

```

@@ -6,8 +6,8 @@ namespace :test do
6   6     task setup: [:environment, 'db:setup', 'db:migrate'] do
7     7       require 'helpers/database_populator'
8     8       dbpop = DatabasePopulator.new
9     -     dbpop.generate_user_roles()
10    9       dbpop.generate_users()
11   10      dbpop.generate_units()
12   11      + dbpop.generate_helpdesk_tickets()
13   12    end
14  end

```

17 test/api/auth_test.rb

[View](#)

```

...
...
@@ -1,6 +1,6 @@
1   1     require 'test_helper'
2   2
3   -class AuthTest < MiniTest::Test
3   +class AuthTest < ActiveSupport::TestCase
4   4       include Rack::Test::Methods
5   5       include TestHelpers::AuthHelper
6   6       include TestHelpers::JsonHelper
7
7 @@ -38,10 +38,7 @@ def test_auth_post
8   38       # These match the model object... so can compare in loops
9   39       user_keys = [ 'id', 'email', 'first_name', 'last_name', 'username', 'nickname', 'receive_task_notifications', 'rece
10  40

```

```
41 -     assert_json_matches_model(response_user_data, expected_auth, user_keys)
42 -
43 -     user_keys.each { |k| assert response_user_data.has_key?(k), "Response has key #{k}" }
44 -     user_keys.each { |k| assert_equal expected_auth[k], response_user_data[k], "Values for key #{k} match" }
45 -     assert_json_matches_model(response_user_data, expected_auth, *user_keys)
46 #
47 # Check other values returned
48 assert_equal expected_auth.name, response_user_data['name'], 'Names match'
49
50 @@ -72,28 +69,28 @@ def test_auth_roles
51     {
52         expect: Role.admin,
53         post: {
54             username: "acain",
55             password: "password"
56         }
57     },
58     {
59         expect: Role.convenor,
60         post: {
61             username: "jrenzella",
62             password: "password"
63         }
64     },
65     {
66         expect: Role.tutor,
67         post: {
68             username: "rwilson",
69             password: "password"
70         }
71     },
72     {
73         expect: Role.student,
74         post: {
75             username: "acummaudo",
76             password: "password"
77         }
78     }
79
80 @@ -116,7 +113,6 @@ def test_auth_roles
81
82     # Test put for authentication token
83     def test_auth_put
84         auth_token = get_auth_token
85         data_to_put = {
86             username: "acain",
87             password: "password"
88         }
89
90     @@ -136,7 +132,6 @@ def test_auth_put
91
92         # Test for deleting authentication token
93         def test_auth_delete
94             auth_token = get_auth_token
95             # Get the auth token needed for delete test
96             delete "/api/auth/#{auth_token}.json", "CONTENT_TYPE" => 'application/json'
97             # 200 response code means success!
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
```

85 test/api/helpdesk/ticket_test.rb

View

... | ... @ -0.0 +1.95 @

```
1 +require 'test_helper'
2 +
3 +class TicketsTest < ActiveSupport::TestCase
4 +  include Rack::Test::Methods
5 +  include TestHelpers::AuthHelper
6 +  include TestHelpers::JsonHelper
7 +
8 +  def app
9 +    Rails.application
10 +  end
11 +
12 +  # POST /helpdesk/tickets
13 +  def test_post_helpdesk_ticket
14 +    project = Randomizer.random_record_for_model(Project)
15 +
16 +    # Can't create ticket if this user already has a ticket
17 +    while HelpdeskTicket.user_has_ticket_open? project.user_id
18 +      project = Randomizer.random_record_for_model(Project)
19 +    end
20 +
21 +    ticket = {
22 +      project_id: project.id,
23 +      description: Populator.words(5..10),
24 +      task_definition_id: rand > 0.33 ? Randomizer.random_task_def_for_project(project).id : nil
25 +    }
26 +
27 +    data_to_post = add_auth_token ticket, project.user
28 +
29 +    post_json '/api/helpdesk/tickets', data_to_post
30 +
31 +    # Last ticket ids should match
32 +    expected_ticket = HelpdeskTicket.last
33 +    assert_json_matches_model last_response_body, expected_ticket, :id
34 +  end
35 +
36 +  # GET /helpdesk/tickets
37 +  def test_get_all_helpdesk_tickets
38 +    get with_auth_token '/api/helpdesk/tickets'
39 +    expect = ActiveModel::ArraySerializer.new(HelpdeskTicket.all, each_serializer: ShallowHelpdeskTicketSerializer)
40 +    assert_json_equal expect, last_response_body
41 +  end
42 +
43 +  # GET /helpdesk/tickets
44 +  def test_get_all_unresolved_helpdesk_tickets
45 +    get with_auth_token '/api/helpdesk/tickets?filter=unresolved'
46 +    expect = ActiveModel::ArraySerializer.new(HelpdeskTicket.all_unresolved, each_serializer: ShallowHelpdeskTicketSerializer)
47 +    assert_json_equal expect, last_response_body
48 +  end
49 +
50 +  # GET /helpdesk/tickets?filter=resolved
51 +  def test_get_resolved_helpdesk_tickets
52 +    get with_auth_token '/api/helpdesk/tickets?filter=resolved'
53 +    expect = ActiveModel::ArraySerializer.new(HelpdeskTicket.all_resolved, each_serializer: ShallowHelpdeskTicketSerializer)
54 +    assert_json_equal expect, last_response_body
55 +  end
56 +
57 +  # GET /helpdesk/tickets?user_id=1&filter=resolved
58 +  def test_get_resolved_helpdesk_tickets_for_user
59 +    id = 1
60 +    get with_auth_token "/api/helpdesk/tickets?user_id=#{id}&filter=resolved"
61 +    expect = ActiveModel::ArraySerializer.new(HelpdeskTicket.all_resolved(id), each_serializer: ShallowHelpdeskTicketSerializer)
62 +    assert_json_equal expect, last_response_body
63 +  end
```

```

64 +
65 + # GET /helpdesk/tickets/:id
66 + # id = 1
67 + def test_get_helpdesk_ticket_with_id
68 +   id = 1
69 +   get with_auth_token "/api/helpdesk/tickets/#{id}"
70 +   expect = HelpdeskTicketSerializer.new HelpdeskTicket.find(id)
71 +   assert_json_equal expect, last_response_body
72 + end
73 +
74 + # DELETE /helpdesk/tickets/:id?resolve=false
75 + def test_delete_a_ticket_by_closing
76 +   ticket = HelpdeskTicket.all_unresolved.first
77 +   delete with_auth_token "/api/helpdesk/tickets/#{ticket.id}"
78 +   # Reload the ticket from DB with updated info
79 +   ticket = HelpdeskTicket.find(ticket.id)
80 +   assert ticket.is_closed
81 +   refute ticket.is_resolved
82 + end
83 +
84 + # DELETE /helpdesk/tickets/:id?resolve=true
85 + def test_delete_a_ticket_by_resolving
86 +   ticket = HelpdeskTicket.all_unresolved.first
87 +   delete with_auth_token "/api/helpdesk/tickets/#{ticket.id}?resolve=true"
88 +   # Reload the ticket from DB with updated info
89 +   ticket = HelpdeskTicket.find(ticket.id)
90 +   assert ticket.is_resolved
91 +   assert ticket.is_closed
92 +   dt = DateTime.parse(last_response_body['closed_at']).inspect
93 +   assert dt, ticket.closed_at
94 + end
95 +end

```

12 test/api/units_test.rb

View
Code

```

...
...
@@ -1,7 +1,7 @@
 1   1 require 'test_helper'
 2   2 require 'date'
 3   3
 4 -class UnitsTest < MiniTest::Test
 5 +class UnitsTest < ActiveSupport::TestCase
 6   include Rack::Test::Methods
 7   include TestHelpers::AuthHelper
 8   include TestHelpers::JsonHelper
 9
@@ -10,10 +10,6 @@ def app
10     Rails.application
11   end
12
13 -  def setup
14 -    @auth_token = get_auth_token()
15 -  end
16
17   # -----
18   # --- Endpoint testing for:
19   # ----- /api/units.json
@@ -100,7 +96,7 @@ def test_units_post_tutorial
100  # Test GET for getting all units
101  def test_units_get
102    # The GET we are testing
103 -    get "/api/units.json?auth_token=#{@auth_token}"
104 +    get with_auth_token "/api/units.json"
105
106    actual_unit = JSON.parse(last_response.body)[0]

```

```

106 102     expected_unit = Unit.first
  @@ -124,7 +120,7 @@ def test_units_get_by_id
124 120         # Get response back from getting a unit by id
125 121
126 122     # Test getting the first unit with id of 1
127 -     get "/api/units/1.json?auth_token=#{@auth_token}"
128 +     get with_auth_token "/api/units/1.json"
129 125
130 126     actual_unit = JSON.parse(last_response.body)
131     expected_unit = Unit.find(1)
  @@ -137,7 +133,7 @@ def test_units_get_by_id
137 133
138 134         # Get response back from getting a unit by id
139 135     # Test getting the first unit with id of 2
140 -     get "/api/units/2.json?auth_token=#{@auth_token}"
141 +     get with_auth_token "/api/units/2.json"
142 137
143 138     actual_unit = JSON.parse(last_response.body)
144     expected_unit = Unit.find(2)
  @@

```

33 test/helpers/auth_helper.rb

View
Code

```

  @@ -4,25 +4,40 @@ module TestHelpers
4   4     #
5   5     module AuthHelper
6   6         #
7 -     # Gets an authentication token for User.first
7 +     # Gets an auth token for the provided user
8   8         #
9 -     def get_auth_token
10 -         user = User.first
11 +     def auth_token(user = User.first)
12         user.extend_authentication_token(true)
13         user.auth_token
14     end
15
16     #
17 -     # Adds an authentication token to the hash of data provided
18 +     # Adds an authentication token to the hash data or string URL
19 +     # This prevents us from having to keep adding the :auth_token
20 +     # key to any POST data that is needed
21 +     # key to any GET/POST/PUT etc. data that is needed
22     #
23 -     def add_auth_token(hash)
24 -         hash[:auth_token] = get_auth_token
25 -         hash
26 +     def add_auth_token(data, user = User.first)
27 +         # Passed in an id instead of a user model? Find the user model from User.find
28 +         user = User.find(user) if user.is_a? Fixnum
29 +         if data.is_a? Hash
30             data[:auth_token] = auth_token user
31         elsif data.is_a? String
32             # If we have a question mark, we need to add a query parameter using &
33             # otherwise use ?
34             data << (data.include?('?') ? "&" : "?") << "auth_token=#{auth_token}"
35         end
36         data
37     end
38
39 -     module_function :get_auth_token
40 +     #

```

```

33 +     # Alias for above for nicer usage (e.g., get with_auth_token "http://")
34 +
35 +     def with_auth_token(data, user = User.first)
36 +       add_auth_token data, user
37 +     end
38 +
39 +     module_function :auth_token
40 +     module_function :add_auth_token
41 +     module_function :with_auth_token
42 +   end
43 end

```

33 test/helpers/json_helper.rb

[View](#)

```

@@ -20,15 +20,40 @@ def put_json(endpoint, data)
20   end
21
22   #
23 - # Assert that a JSON response matches the model and keys provided
24 + # Assert that a JSON response matches the model and keys provided.
25 + # Keys MUST be symbols
26   #
27 - def assert_json_matches_model(response_json, model, keys)
28 -   keys.each { |k| assert response_json.has_key?(k), "Response has key #{k}"}
29 -   keys.each { |k| assert_equal model[k], response_json[k], "Values for key #{k} match" }
30 + def assert_json_matches_model(response_json, model, *keys)
31 +   keys.each { |k| assert response_json.key?(k.to_s), "Response has key #{k}"}
32 +   keys.each { |k| assert_equal model[k], response_json[k.to_s], "Values for key #{k} match" }
33 + end
34 +
35 + #
36 + # Last response body parsed from JSON
37 + #
38 + def last_response_body
39 +   JSON.parse(last_response.body)
40 + end
41 +
42 + #
43 + # Converts from an ActiveRelation to JSON (without Ruby objects inside the hash)
44 + #
45 + def json_hashed(hash)
46 +   JSON.parse(hash.to_json)
47 + end
48 +
49 + #
50 + # Assert that the lefthand matches the right-hand as json hash
51 + #
52 + def assert_json_equal(lhs, rhs)
53 +   assert_equal json_hashed(lhs), json_hashed(rhs)
54 + end
55 +
56 + module_function :assert_json_matches_model
57 + module_function :post_json
58 + module_function :put_json
59 + module_function :last_response_body
60 + module_function :json_hashed
61 + module_function :assert_json_equal
62 end
63 end

```

68 test/models/helpdesk/ticket_test.rb

[View](#)

*** *** @@ -0,0 +1,68 @@

```
1 +require "test_helper"
2 +
3 +class HelpdeskTicketTest < ActiveSupport::TestCase
4 +  setup do
5 +    @test_desc = 'Hello, World!'
6 +    @test_project = Randomizer.random_record_for_model(Project)
7 +    @test_task = Randomizer.random_task_for_project(@test_project)
8 +    @tickets = {
9 +        without_task: HelpdeskTicket.create!(
10 +            project: @test_project
11 +        ),
12 +        with_task: HelpdeskTicket.create!(
13 +            project: @test_project,
14 +            task: @test_task
15 +        ),
16 +        with_description: HelpdeskTicket.create!(
17 +            project: @test_project,
18 +            description: @test_desc
19 +        )
20 +    }
21 +  end
22 +
23 +  test "helpdesk tickets have projects" do
24 +    @tickets.values.each do |t|
25 +      assert t.project, @test_project
26 +    end
27 +  end
28 +
29 +  test "only one helpdesk ticket has a description" do
30 +    assert_not_nil @tickets[:with_description].description
31 +    assert_equal @tickets[:with_description].description, @test_desc
32 +    assert_nil @tickets[:with_task].description
33 +    assert_nil @tickets[:without_task].description
34 +  end
35 +
36 +  test "only one helpdesk ticket has a task" do
37 +    assert_equal @tickets[:with_task].task, @test_task
38 +    assert_nil @tickets[:with_description].task
39 +    assert_nil @tickets[:without_task].task
40 +  end
41 +
42 +  test "resolving a helpdesk ticket" do
43 +    ticket_to_resolve = @tickets[:without_task]
44 +    ticket_to_resolve.resolve
45 +    assert ticket_to_resolve, true
46 +    # both without_task and ticket_to_resolve should match
47 +    assert_equal @tickets[:without_task], ticket_to_resolve
48 +    # others should remain false
49 +    refute @tickets[:with_task].is_resolved
50 +    refute @tickets[:with_description].is_resolved
51 +  end
52 +
53 +  test "the task? method should be true if the task has a task" do
54 +    assert @tickets[:with_task].task?
55 +  end
56 +
57 +  test "the student method should match the associated project's student" do
58 +    assert_equal @tickets[:with_task].student, @test_project.student
59 +    assert_equal @tickets[:with_task].student, @test_project.student
60 +    assert_equal @tickets[:with_description].student, @test_project.student
61 +  end
62 +
63 +  test "the unit method should match the associated project's unit" do
```

```

64 +   assert_equal @tickets[:with_task].unit, @test_project.unit
65 +   assert_equal @tickets[:with_task].unit, @test_project.unit
66 +   assert_equal @tickets[:with_description].unit, @test_project.unit
67 + end
68 +end

```

18 test/test_helper.rb

[View](#)

```

@@ -17,16 +17,14 @@
17 # Require minitest extensions
18 require 'minitest/rails'
19 require 'minitest/pride'
20 -require 'minitest/autorun'
21 20 require 'minitest/osx'
22 +require 'minitest/around'

23 -# Require all test helpers
24 -require_all 'test/helpers'
25 23 +# Require test helpers
26 24 require 'rails/test_help'
27 25 require 'database_cleaner'

28 class ActiveSupport::TestCase
29 - # Check if migrations are pending
30 28   ActiveRecord::Migration.check_pending!

31 29

32 30   # Setup all fixtures in test/fixtures/*.yml|csv for all tests in alphabetical order.
@@ -41,17 +39,17 @@ class ActiveSupport::TestCase
39   # Support rollback of db changes after all tests
40   DatabaseCleaner.strategy = :transaction
41 41

44 - # After setup of all test, start database cleaner to undo transactions
45 - def before_teardown
46 -   super
47 42 + def setup
48 43   DatabaseCleaner.start
49 44 end
50 45

51 - # After teardown
52 - def after_teardown
53 46 + def teardown
54 47   DatabaseCleaner.clean
55 48   super
56 49

50 50   # Add more helper methods to be used by all tests here...
51 + require_all 'test/helpers'
52 +
53 + extend MiniTest::Spec::DSL
54 +
55 55 end

```

4 Doubtfire Web Source Code Changes

See attached document

2 Gruntfile.js

View

```

@@ -358,7 +358,7 @@ module.exports = function ( grunt ) {
 358     sub: true,
 359     boss: true,
 360     eqnull: true
 361   },
 361 }
 362 // No test or globals provided - comment out to silence warnings
 363 // test: [
 364   // '<%= app_files.jsunit %>'
```

68 src/app/api/models/helpdesk-session.coffee

View

```

...
...
@@ -0,0 +1,68 @@
1 +angular.module("doubtfire.api.models.helpdesk-session", [])
2 +#
3 +## API endpoint for helpdesk sessions of staff working
4 +#
5 +.factory("HelpdeskSession", (resourcePlus) ->
6 +  HelpdeskSession = resourcePlus "/helpdesk/sessions/:id", { id: "@id" }
7 +
8 +  #
9 +  # Clocks on a user for the specified duration (in hours) given. Durations
10 + # must be specified in decimal format. E.g., 1h 15m == 1.25 hrs
11 + #
12 + HelpdeskSession.clockOn = (userId, duration, callback) ->
13 +  clockOffTime = moment().add(duration, 'hours').format()
14 +  dataToPost =
15 +    clock_off_time: clockOffTime
16 +    user_id: userId
17 +  onSuccess = (response) -> callback(null, response)
18 +  onFailure = (response) -> callback(response)
19 +  HelpdeskSession.create(dataToPost).$promise.then(onSuccess, onFailure)
20 +
21 + #
22 + # Gets the active working session of a specific userId at the helpdesk
23 + #
24 + HelpdeskSession.getActiveSessions = (userId, callback) ->
25 +  onSuccess = (response) -> callback(null, response)
26 +  onFailure = (response) -> callback(response)
27 +  HelpdeskSession.query({ user_id: userId, is_active: true }).$promise.then(onSuccess, onFailure)
28 +
29 + #
30 + # Gets the list of all tutors currently working
31 + #
32 + HelpdeskSession.tutorsWorkingNow = (callback) ->
33 +  onSuccess = (response) -> callback(null, response)
34 +  onFailure = (response) -> callback(response)
35 +  HelpdeskSession.query({ is_active: true }).$promise.then(onSuccess, onFailure)
36 +
37 + #
38 + # Prematurely clocks off a session before it's default expiry time
39 + #
40 + HelpdeskSession.clockOffSession = (sessionId, callback) ->
41 +  onSuccess = (response) -> callback(null, response)
42 +  onFailure = (response) -> callback(response)
43 +  HelpdeskSession.delete({ id: sessionId }).$promise.then(onSuccess, onFailure)
44 +
45 + #
46 + # Clocks off this session prematurely
```

```

47 + #
48 + HelpdeskSession.prototype.clockOff = (callback) ->
49 +   HelpdeskSession.clockOffSession(this.id, callback)
50 +
51 +
52 + # Returns the duration until the session expires
53 + #
54 + HelpdeskSession.prototype.timeUntilFinish = ->
55 +   clockOffTime = moment(this.clock_off_time)
56 +   moment.duration(clockOffTime.diff())
57 +
58 + #
59 + # Returns the currently working session for the user id provided, or
60 + # null if no such session exists
61 + #
62 + HelpdeskSession.currentWorkingSession = (userId, callback) ->
63 +   HelpdeskSession.getActiveSessions userId, (error, success) ->
64 +     return callback(error) if error
65 +     callback(null, success[0])
66 +
67 + HelpdeskSession
68 +)

```

21 src/app/api/models/helpdesk-stats.coffee

[View](#)

```

...
...
@@ -0,0 +1,21 @@
1 +angular.module("doubtfire.api.models.helpdesk-stats", [])
2 +#
3 +## API endpoint for getting helpdesk stats
4 +#
5 +.factory("HelpdeskStats", ($http, $rootScope, $interval, api) ->
6 +  HelpdeskStats = {}
7 +
8 + #
9 + # Gets the helpdesk statistics from the time specified by `from` to the
10 + # time specified by `to`.
11 + #
12 + # If to is null, statistics will fetch to now.
13 + # If from is null, all statistics from the start of time are returned.
14 + #
15 + HelpdeskStats.get = (from, to, interval, callback) ->
16 +   onSuccess = (response) -> callback(null, response.data)
17 +   onFailure = (response) -> callback(response)
18 +   $http.get("#{api}/helpdesk/stats/dashgraph").then(onSuccess, onFailure)
19 +
20 + HelpdeskStats
21 +)

```

103 src/app/api/models/helpdesk-ticket.coffee

[View](#)

```

...
...
@@ -0,0 +1,103 @@
1 +angular.module("doubtfire.api.models.helpdesk-ticket", [])
2 +#
3 +## API endpoint for serving helpdesk tickets
4 +#
5 +.factory("HelpdeskTicket", (resourcePlus) ->
6 +  HelpdeskTicket = resourcePlus "/helpdesk/tickets/:id", { id: "@id" }
7 +
8 + #
9 + # Submits a new helpdesk ticket. Provide a project ID, description and task
10 + # def id. Project ID is required, whereas the others can be `null`. Callback
11 + # is a function whose first parameter is error and second parameter is success
12 + # data.

```

```
13 + #
14 + HelpdeskTicket.submitTicket = (projectId, description, taskDefinitionId, callback) ->
15 +   dataToPost =
16 +     project_id: projectId
17 +     description: description
18 +     task_definition_id: taskDefinitionId
19 +   onSuccess = (response) -> callback(null, response)
20 +   onFailure = (response) -> callback(response)
21 +   HelpdeskTicket.create(dataToPost)
22 +     .$promise
23 +     .then(onSuccess, onFailure)
24 +
25 + #
26 + # Gets all tickets by a specific state, optionally for the specified user id.
27 + # If user is not specified, then all tickets will be returned.
28 + #
29 + HelpdeskTicket.getTicketsByState = (state, userId, callback) ->
30 +   onSuccess = (response) -> callback(null, response)
31 +   onFailure = (response) -> callback(response)
32 +   HelpdeskTicket.query({ user_id: userId, shallow: userId?, filter: state })
33 +     .$promise
34 +     .then(onSuccess, onFailure)
35 +
36 + #
37 + # Gets all unresolved tickets (optionally by user)
38 + #
39 + HelpdeskTicket.getUnresolvedTickets = (userId, callback) ->
40 +   HelpdeskTicket.getTicketsByState('unresolved', userId, callback)
41 +
42 + #
43 + # Gets all resolved tickets (optionally by user)
44 + #
45 + HelpdeskTicket.getResolvedTickets = (userId, callback) ->
46 +   HelpdeskTicket.getTicketsByState('resolved', userId, callback)
47 +
48 + #
49 + # Gets all closed tickets (optionally by user)
50 + #
51 + HelpdeskTicket.getClosedTickets = (userId, callback) ->
52 +   HelpdeskTicket.getTicketsByState('closed', userId, callback)
53 +
54 + #
55 + # Closes a specified ticket with the given ID
56 + #
57 + HelpdeskTicket.closeTicket = (ticketId, callback) ->
58 +   onSuccess = (response) -> callback(null, response)
59 +   onFailure = (response) -> callback(response)
60 +   HelpdeskTicket.delete({ id: ticketId })
61 +     .$promise
62 +     .then(onSuccess, onFailure)
63 +
64 + #
65 + # Prototype close ticket method
66 + #
67 + HelpdeskTicket.prototype.close = (callback) ->
68 +   HelpdeskTicket.closeTicket(this.id, callback)
69 +
70 + #
71 + # Resolves a specified ticket with the given ID
72 + #
73 + HelpdeskTicket.resolveTicket = (ticketId, callback) ->
74 +   onSuccess = (response) -> callback(null, response)
75 +   onFailure = (response) -> callback(response)
```

```

76 +     HelpdeskTicket.delete({ id: ticketId, resolve: true })
77 +         .$promise
78 +             .then(onSuccess, onFailure)
79 +
80 + #
81 + # Prototype resolve ticket method
82 + #
83 + HelpdeskTicket.prototype.resolve = (callback) ->
84 +     HelpdeskTicket.resolveTicket(this.id, callback)
85 +
86 + #
87 + # Returns the duration since the ticket was open
88 + #
89 + HelpdeskTicket.prototype.lengthOfTimeOpen = ->
90 +     createdTime = moment(this.created_at)
91 +     moment.duration(createdTime.diff())
92 +
93 + #
94 + # Checks if the user with the specified ticket has a ticket open already
95 + # and if so returns that ticket in the callback
96 + #
97 + HelpdeskTicket.currentOpenTicket = (userId, callback) ->
98 +     HelpdeskTicket.getUnresolvedTickets userId, (error, success) ->
99 +         return callback(error) if error
100 +         callback(null, success[0])
101 +
102 +     HelpdeskTicket
103 +

```

3 src/app/api/models/models.coffee

[View](#) 

```

@@ -8,6 +8,9 @@ angular.module("doubtfire.api.models", [
8   8     "doubtfire.api.models.tutorial"
9   9     "doubtfire.api.models.learning-alignments"
10 10     "doubtfire.api.models.intended-learning-outcome"
11 +    "doubtfire.api.models.helpdesk-ticket"
12 +    "doubtfire.api.models.helpdesk-session"
13 +    "doubtfire.api.models.helpdesk-stats"
11 14     "doubtfire.api.models.task"
12 15     "doubtfire.api.models.task-comment"
13 16     "doubtfire.api.models.task-definition"

```

1 src/app/app.coffee

[View](#) 

```

@@ -16,6 +16,7 @@ angular.module('doubtfire', [
16 16     'doubtfire.tasks'
17 17     'doubtfire.projects'
18 18     'doubtfire.users'
19 +    'doubtfire.helpdesk'
20     'doubtfire.groups'
21     'doubtfire.visualisations'
22 ])

```

4 src/app/common/header/header.coffee

[View](#) 

```

... ...
@@ -1,7 +1,9 @@
1   1 #
2   2 # Controllers and providers related to the header/nav bar
3   3 #
4 -angular.module('doubtfire.common.header', [])
4 +angular.module('doubtfire.common.header', [
5 +    'doubtfire.common.header.helpdesk-header'

```

```
6 | +])  
5 |  
6 | .controller("BasicHeaderCtrl", ($scope, $state, $modal, User, AboutDoubtfireModal, UserNotificationSettingsModal, UserS  
7 |   $scope.menus = headerService.getMenus()
```

30 src/app/common/header/header.scss

View

```
@@ -5,6 +5,16 @@
  border-right: none;
  margin-bottom: 0;

+ // Mobile responsive fixes
+ @media (min-width: $screen-md) {
+   // Fix for dynamic content
+   .navbar-right {
+     width: 80%;
+     display: flex;
+     justify-content: flex-end;
+   }
+ }

.dropdown-units {
  li:not(.divider) a {
    padding-top: 0.5em;
}

@@ -18,6 +28,8 @@
  padding-left: 1.75ex;
  flex: 1;
  text-align: right;
+ overflow: hidden;
+ text-overflow: ellipsis;
}

}

@@ -35,3 +47,21 @@
  a.navbar-brand {
  margin: 0;
}

+
+@media (max-width: $screen-md) {
+ .navbar {
+   position: fixed;
+   top: 0;
+   left: 0;
+   right: 0;
+   z-index: 9;
+ }
+ .navbar .navbar-nav .open ul.dropdown-menu {
+   background-color: #fcfcfc;
+   border-bottom: 1px solid #aaa;
+   border-radius: 0;
+ }
+ body {
+   padding-top: $navbar-height;
+ }
+}
```

13 src/app/common/header/header.tpl.html

View

```
  @@ -15,15 +15,10 @@ <h1 class="logo">Doubtfire</h1>
15    15    <!-- Collect the nav links, forms, and other content for toggling -->
16    16    <div class="collapse navbar-collapse" ng-class="!navCollapsed && 'in'" ng-show="!navCollapsed">
```

```

17      <ul class="nav navbar-nav navbar-right">
18      -<li class="dropdown dropdown-admin" dropdown ng-repeat="menu in menus | orderBy:'name'">
19          -<a class="dropdown-toggle" dropdown-toggle role="button"><span class="glyphicon glyphicon-{{menu.icon}}><{{menu
20              -<ul class="dropdown-menu">
21                  -<li ng-repeat="link in menu.links | orderBy:'name'">
22                      -<ng-class="{active: link.class === 'active'}><a href="{{link.url}}>{{link.name}}</a>
23                  -</li>
24              -</ul>
25          -</li>
26      -</ul>
27      +<helpdesk-header
28          + projects="projects"
29          + unit-roles="unitRoles">
27      +</helpdesk-header>
28
29      <li class="dropdown dropdown-units" dropdown ng-hide="projects.length == 0 && unitRoles.length == 0">
30          <a role="button" class="dropdown-toggle" dropdown-toggle><span class="glyphicon glyphicon-book"></span> Units <
31

```

57 src/app/common/header/helpdesk-header/helpdesk-header.coffee

[View](#)

```

...
...
@@ -0,0 +1,57 @@
1  +#
2  ## Directive for controlling helpdesk dropdown
3  +#
4  +angular.module('doubtfire.common.header.helpdesk-header', [])
5  +.directive 'helpdeskHeader', -
6  +  restrict: 'E'
7  +  replace: true
8  +  scope:
9  +    projects: '='
10 +   unitRoles: '='
11 +   templateUrl: 'common/header/helpdesk-header/helpdesk-header.tpl.html'
12 +   controller: ($scope, $rootScope, currentUser, HelpdeskTicketModal, HelpdeskSessionModal, HelpdeskTicket, HelpdeskSession) =>
13 +     #
14 +     # This function updates the visibility of the element
15 +     #
16 +     updateVisibility = ->
17 +       # These variables help us know if the user teaches or studies a unit
18 +       $scope.studiesUnit = $scope.projects?.length > 0
19 +       $scope.teachesUnit = $scope.unitRoles?.length > 0
20 +       # Only show the helpdesk dropdown if I'm learning or teaching a unit
21 +       $scope.showHelpdeskDropdown = $scope.studiesUnit or $scope.teachesUnit
22 +       # Check if the user has a ticket/session open to switch which modals they can open
23 +       if $scope.studiesUnit
24 +         HelpdeskTicket.currentOpenTicket $scope.currentUser.id, (error, data) ->
25 +           $scope.currentOpenTicket = data
26 +       if $scope.teachesUnit
27 +         HelpdeskSession.currentWorkingSession $scope.currentUser.id, (error, data) ->
28 +           $scope.currentWorkingSession = data
29 +
30 +
31 +     #
32 +     # Watch for changes in projects and unitRoles length's and update the
33 +     # visibility in case they change
34 +     #
35 +     $scope.$watch('projects.length', updateVisibility)
36 +     $scope.$watch('unitRoles.length', updateVisibility)
37 +
38 +     # Current user binding
39 +     $scope.currentUser = currentUser.profile
40 +
41 +     # Watch the root scope for changes to the current ticket/session and set if needed

```

```

42 + $rootScope.$on 'CurrentOpenTicket', (event, ticket) ->
43 +   $scope.currentOpenTicket = ticket
44 + $rootScope.$on 'CurrentWorkingSession', (event, session) ->
45 +   $scope.currentWorkingSession = session
46 +
47 + #
48 + # Opens the submit ticket modal
49 + #
50 + $scope.openHelpdeskTicketModal = ->
51 +   HelpdeskTicketModal.show $scope.currentOpenTicket
52 +
53 + #
54 + # Opens the session modal
55 + #
56 + $scope.openHelpdeskSessionModal = ->
57 +   HelpdeskSessionModal.show $scope.currentWorkingSession

```

20 src/app/common/header/helpdesk-header/helpdesk-header.scss

[View](#)

```

... ...
@@ -0,0 +1,20 @@
1 // Mobile responsive fixes
2 +@media (min-width: $screen-md) {
3 + .helpdesk.header {
4 +   display: inline;
5 +   .helpdesk-direct-link {
6 +     padding-right: 0;
7 +     .badge {
8 +       background-color: $doubtfire-color;
9 +     }
10 +   }
11 +   .dropdown-toggle {
12 +     float: right;
13 +   }
14 + }
15 +
16 +@media (max-width: $screen-md) {
17 + .helpdesk.header .helpdesk-direct-link {
18 +   display: none;
19 + }
20 +}

```

33 src/app/common/header/helpdesk-header/helpdesk-header.tpl.html

[View](#)

```

... ...
@@ -0,0 +1,33 @@
1 +<li dropdown class="helpdesk header" ng-show="showHelpdeskDropdown">
2 +  <a class="helpdesk-direct-link pull-left" ng-click="openHelpdeskTicketModal()" ng-if="currentOpenTicket">
3 +    <span class="badge" tooltip="Ticket open for {{currentOpenTicket.lengthOfTimeOpen().humanize()}}" tooltip-placement="top">
4 +      <i class="fa fa-ticket"></i>
5 +    </span>
6 +  </a>
7 +  <a class="helpdesk-direct-link pull-left" ng-click="openHelpdeskSessionModal()" ng-if="currentWorkingSession">
8 +    <span class="badge" tooltip="You finish working in {{currentWorkingSession.timeUntilFinish().humanize()}}" tooltip-placement="top">
9 +      <i class="fa fa-clock-o"></i>
10 +    </span>
11 +  </a>
12 +  <a class="dropdown-toggle" dropdown-toggle>
13 +    <i class="fa fa-ambulance"></i>
14 +  Helpdesk
15 +  <b class="caret"></b>
16 + </a>
17 + <ul class="dropdown-menu">
18 +   <li ng-if="studiesUnit">
19 +     <a ng-click="openHelpdeskTicketModal()">

```

```

20      {{ currentOpenTicket ? "View Open Ticket" : "Submit a Ticket" }}
21    </a>
22  </li>
23  <li ng-if="teachesUnit">
24    <a ng-click="openHelpdeskSessionModal()">
25      {{ currentWorkingSession ? "Clock Off" : "Clock On" }}
26    </a>
27  </li>
28  <li class="divider"></li>
29  <li>
30    <a ui-sref="helpdesk"> View Helpdesk Dashboard </a>
31  </li>
32 </ul>
33 +</li>

```

20 src/app/common/modals/confirmation-modal/confirmation-modal.tpl.html

View
Code

```

@@ -6,17 +6,17 @@ <h3 class="modal-title">
  <small>Please confirm that you want to perform this action.</small>
</h3>
</div>
<div class="modal-body">
  {{message}}
<div class="modal-body clearfix">
  <p class="col-sm-12">
    {{message}}
  </p>
</div>
<div class="modal-footer clearfix">
  <div class="btn-group">
    <label class="btn btn-success" ng-click="confirmAction()">
      <i class="fa fa-check"></i> Confirm
    </label>
    <label class="btn btn-danger" ng-click="cancelAction()">
      <i class="fa fa-times"></i> Cancel
    </label>
  </div>
  <button class="btn btn-danger" ng-click="cancelAction()">
    <i class="fa fa-times"></i> Cancel
  </button>
  <button class="btn btn-success" ng-click="confirmAction()">
    <i class="fa fa-check"></i> Confirm
  </button>
</div>
</div>

```

10 src/app/common/services/project-service.coffee

View
Code

```

@@ -26,7 +26,7 @@ angular.module("doubtfire.common.services.projects", [])
  projectService.loadedProjects = null

  projectService.getProjects = ( callback ) =>
-  if ! projectService.loadedProjects?
+  unless projectService.loadedProjects?
    projectService.loadedProjects = []
    Project.query(
      (projects) =>
@@ -43,6 +43,14 @@ angular.module("doubtfire.common.services.projects", [])
  projectService.loadedProjects

  +
  #+
  # Finds a project with the given id

```

```

48 +  #
49 +  projectService.findProject = (id, callback) ->
50 +    projectService.getProjects (projects) ->
51 +      projFound = _.find(projects, (p) -> p.project_id is id)
52 +      callback(projFound)
53 +
54  #####
55  projects's can update their task stats
56  converts the | delimited stats to its component arrays

```

2 src/app/common/services/unit-service.coffee

[View](#)


```

@@ -193,7 +193,7 @@ angular.module("doubtfire.common.services.units", [])
193   193     (response) ->
194   194       alertService.add("danger", "Grade was not updated: #{response.data.error}", 8000)
195   195
196 -    projectService.updateTaskStats(student, student.stats)
196 +    projectService.updateTaskStats(student, student.stats) if student.stats
197   197     projectService.addTaskDetailsToProject(student, unit)
198   198
199   199     unit.getGroups = (group_set, group_callback) ->

```

33 src/app/helpdesk/helpdesk-ticket/helpdesk-ticket.coffee

[View](#)


```

... ...
1  @@ -0,0 +1,33 @@
2  +angular.module('doubtfire.helpdesk.helpdesk-ticket', [])
3  +#
4  +## Directive to display information related to a helpdesk ticket
5  +#
6  +.directive('helpdeskTicket', ->
7  +  restrict: 'E'
8  +  templateUrl: 'helpdesk/helpdesk-ticket/helpdesk-ticket.tpl.html'
9  +  replace: true
10 +  scope:
11 +    ticket: '='
12 +  controller: ($scope, HelpdeskTicket, projectService, unitService, HelpdeskTicketModal) ->
13 +
14 +    #
15 +    # Returns warning style for ticket
16 +    #
17 +    $scope.warningStyle = ->
18 +      minsWaiting = -$scope.ticket.lengthOfTimeOpen().asMinutes()
19 +      if minsWaiting < 3
20 +        'success'
21 +      else if 3 <= minsWaiting < 6
22 +        'primary'
23 +      else if 6 <= minsWaiting < 9
24 +        'warning'
25 +      else if minsWaiting >= 9
26 +        'danger'
27 +
28 +    #
29 +    # Opens the submit ticket modal
30 +    #
31 +    $scope.openHelpdeskTicketModal = ->
32 +      return if $scope.isLoading
33 +      HelpdeskTicketModal.show $scope.ticket

```

49 src/app/helpdesk/helpdesk-ticket/helpdesk-ticket.scss

[View](#)


```

...
@@ -0,0 +1,49 @@
1+.helpdesk-ticket {
2+ $br: 8px;
3+ font-size: 1.3em;
4+ // Mobile responsive fixes
5+ @media (max-width: $screen-md) {
6+   margin-bottom: 1em;
7+   .unit, .unit-name, .task-name {
8+     text-align: center !important;
9+     width: 100% !important;
10+ }
11+ }
12+ header, .content {
13+   padding: 8px;
14+ }
15+ header {
16+   background-color: $doubtfire-color;
17+   color: #fff;
18+   font-size: 1.25em;
19+   border-radius: $br $br 0 0;
20+   line-height: 1.25;
21+ }
22+ .content {
23+   color: #525252;
24+   background-color: #eecedf;
25+   .unit-name {
26+     text-overflow: ellipsis;
27+     width: 65%;
28+     text-align: right;
29+     overflow: hidden;
30+     white-space: nowrap;
31+   }
32+ }
33+ .content .loading {
34+   text-align: center;
35+   font-size: 2em;
36+ }
37+ hr {
38+   border: 1px dashed grey;
39+ }
40+ border: 2px solid #eecedf;
41+ border-radius: $br;
42+ transition: all 0.25s ease;
43+ &:hover {
44+   position: relative;
45+   top: -10px;
46+   cursor: pointer;
47+   box-shadow: 0px 0px 10px 0px rgba(0,0,0,0.4);
48+ }
49+}

```

29 src/app/helpdesk/helpdesk-ticket/helpdesk-ticket.tpl.html

[View](#)

```

...
@@ -0,0 +1,29 @@
1+<div class="helpdesk-ticket" ng-click="openHelpdeskTicketModal()">
2+ <header ng-if="isLoading">
3+   <strong>Doubtfire</strong> Helpdesk Ticket
4+ </header><!--/ticket-header-->
5+ <header ng-if="!isLoading">
6+   <strong>{{ticket.project.student_name}}</strong> <span class="pull-right">#{{ticket.position}}</span>
7+ </header><!--/ticket-header-->
8+ <div class="content">
9+   <div class="loading" ng-if="isLoading">
```

```

10 +      <i class="fa fa-spinner fa-pulse"></i>
11 +    </div><!--/loading-ticket-->
12 +    <div ng-if="!isLoading">
13 +      <div class="unit">
14 +        <label class="label label-info">{{ticket.project.unit_code}}</label>
15 +        <strong class="unit-name pull-right">{{ticket.project.unit_name}}</strong>
16 +      </div>
17 +      <div class="task-name text-right">
18 +        {{ticket.task_definition.name}}
19 +      </div>
20 +      <hr>
21 +      <div class="text-center">
22 +        Waiting for
23 +        <strong class="text-{{warningStyle()}}">
24 +          {{ticket.lengthOfTimeOpen().humanize()}}
25 +        </strong>
26 +      </div>
27 +    </div><!--/data-loaded-->
28 +  </div><!--/ticket-content-->
29 +</div>

```

5 src/app/helpdesk/helpdesk.coffee

[View](#)

```

... ...
@@ -0,0 +1,5 @@
1 +angular.module('doubtfire.helpdesk', [
2 +  'doubtfire.helpdesk.modals'
3 +  'doubtfire.helpdesk.states'
4 +  'doubtfire.helpdesk.helpdesk-ticket'
5 +])

```

4 src/app/helpdesk/modals/modals.coffee

[View](#)

```

... ...
@@ -0,0 +1,4 @@
1 +angular.module('doubtfire.helpdesk.modals', [
2 +  'doubtfire.helpdesk.modals.ticket-modal'
3 +  'doubtfire.helpdesk.modals.session-modal'
4 +])

```

81 src/app/helpdesk/modals/session-modal/session-modal.coffee

[View](#)

```

... ...
@@ -0,0 +1,81 @@
1 +#
2 +## The helpdesk session modal is the modal by which staff can clock on and clock
3 +## off from the helpdesk
4 +#
5 +angular.module('doubtfire.helpdesk.modals.session-modal', [])
6 +
7 +.factory('HelpdeskSessionModal', ($modal) =>
8 +  HelpdeskSessionModal = {}
9 +
10 +  #
11 +  # Pass in the user, project
12 +  #
13 +  HelpdeskSessionModal.show = (session) =>
14 +    $modal.open(
15 +      templateUrl: 'helpdesk/modals/session-modal/session-modal.tpl.html'
16 +      controller: 'HelpdeskSessionModal'
17 +      resolve:
18 +        session: => session
19 +
20 +  HelpdeskSessionModal
21 +)
22 +

```

```

23+.controller('HelpdeskSessionModal', ($scope, $state, $rootScope, $interval, $modalInstance, HelpdeskSession, ConfirmationModal) =>
24+  $scope.clockOffState = session? # I can clock off if I was given a session
25+  $scope.clockOnState = !session? # I can clock on if I wasn't given a session
26+
27+  # Strip the profile
28+  $scope.currentUser = currentUser.profile
29+
30+  # Duration to clock on
31+  $scope.duration = { hours: null }
32+
33+  #
34+  # Updates the automatic clock off time
35+  #
36+  updateAutomaticClockOffTime = (newDuration) ->
37+    return unless newDuration?
38+    $scope.automaticClockOffTime =
39+      moment().add($scope.duration.hours, 'hours')
40+
41+  # Watch duration to calculate the automaticClockOffTime
42+  $scope.$watch 'duration.hours', updateAutomaticClockOffTime
43+
44+  # If we have a session, work out the automatic clock off time
45+  if $scope.clockOffState
46+    $scope.automaticClockOffTime = moment(session.clock_off_time)
47+    updateTimeLeft = ->
48+      $scope.timeLeft = moment.duration($scope.automaticClockOffTime.diff())
49+    updateTimeLeft()
50+    updateTimeLeftInterval = $interval updateTimeLeft, 1000
51+    # Must explicitly destroy the time interval when the controller ends!
52+    $scope.$on '$destroy', ->
53+      $interval.cancel(updateTimeLeftInterval)
54+
55+  #
56+  # Clocks on the session
57+  #
58+  $scope.clockOn = ->
59+    clockOnCallback = (error, success) ->
60+      if success
61+        $modalInstance.close(success)
62+        $rootScope.$broadcast('CurrentWorkingSession', success)
63+        alertService.add("success", "Clocked on successfully.", 2000)
64+      if error
65+        alertService.add("danger", "Error: #{error.data.error}", 6000)
66+    HelpdeskSession.clockOn(currentUser.id, $scope.duration.hours, clockOnCallback)
67+
68+  #
69+  # Clocks off the session
70+  #
71+  $scope.clockOff = ->
72+    clockOffCallback = (error, success) ->
73+      if success
74+        $modalInstance.close(success)
75+        $rootScope.$broadcast('CurrentWorkingSession', null)
76+        alertService.add("success", "You have been clocked off.", 2000)
77+      if error
78+        alertService.add("danger", "Error: #{error.data.error}", 6000)
79+    ConfirmationModal.show 'Clock off now', "You still have #{session.timeUntilFinish().humanize()} left to work. Are you sure you want to clock off?"
80+    session.clockOff(clockOffCallback)
81+)

```

8 src/app/helpdesk/modals/session-modal/session-modal.scss

[View](#)

... ... @@ -0,0 +1,8 @@

```

1+.session-modal {
2+ .duration-entry {
3+   margin: 1em auto;
4+ }
5+ .time-left {
6+   font-size: 3em;
7+ }
8+}

```

62 src/app/helpdesk/modals/session-modal/session-modal.tpl.html

[View](#)

```

...
... @@ -0,0 +1,62 @@
1+<div class="session-modal">
2+ <form class="form-horizontal" name="sessionForm" role="form">
3+   <div class="modal-header">
4+     <h3>Clock {{clockOnState ? 'on to' : 'off from'}} the helpdesk</h3>
5+   </div><!--/new-ticket-header-->
6+   <div class="modal-body text-center clearfix">
7+     <div class="col-sm-12" ng-if="clockOnState">
8+       <label>
9+         Please enter how long you are working at the helpdesk for.
10+      </label>
11+      <div class="input-group duration-entry col-sm-4">
12+        <input name="duration"
13+          ng-model="duration.hours"
14+          class="form-control input-lg"
15+          type="number"
16+          placeholder="2"
17+          step=".25"
18+          min="0.25"
19+          max="8"
20+          required>
21+        <span class="input-group-addon">hours</span>
22+      </div>
23+      <p class="text-danger" ng-show="sessionForm.duration.$error.number">
24+        Please enter duration as a decimal number. For example, 1 hour and
25+        15 minutes would be represented as <code>1.25</code> hours.
26+      </p>
27+      <p class="text-danger" ng-show="sessionForm.duration.$error.min">
28+        You must work for at least fifteen minutes.
29+      </p>
30+      <p class="text-danger" ng-show="sessionForm.duration.$error.max">
31+        You can't work more than eight hours.
32+      </p>
33+      <p class="help-block" ng-show="sessionForm.duration.$valid">
34+        You will be automatically clocked off at <strong>{{automaticClockOffTime.format('hh:mm a')}}</strong>,
35+        but you can manually clock off at an earlier time.
36+      </p>
37+    </div><!--/clock-on-state-->
38+    <div class="col-sm-12" ng-if="clockOffState">
39+      <label>
40+        You finish working at the helpdesk in
41+      </label>
42+      <div class="time-left">
43+        {{timeLeft.hours()}}h {{timeLeft.minutes()}}m {{timeLeft.seconds()}}s
44+      </div>
45+      <p class="help-block">
46+        (That's at {{automaticClockOffTime.format('hh:mm a')}})
47+      </p>
48+    </div>
49+  </div>
50+  <div class="modal-footer text-right">
51+    <button ng-click="$dismiss()" class="btn btn-primary">

```

```

52     +     Close
53     +   </button>
54     +   <button ng-disabled="!sessionForm.$valid" ng-click="clockOn()" class="btn btn-success" ng-if="clockOnState">
55     +     <i class="fa fa-clock-o"></i> <i class="fa fa-sign-in"></i> Clock On
56     +   </button>
57     +   <button ng-click="clockOff()" class="btn btn-danger" ng-if="clockOffState">
58     +     <i class="fa fa-clock-o"></i> <i class="fa fa-sign-out"></i> Clock Off Now
59     +   </button>
60   + </div>
61 + </form>
62 +</div>

```

107 src/app/helpdesk/modals/ticket-modal/ticket-modal.coffee

[View](#)

```

...
@@ -0,0 +1,107 @@
+
+#
+## The helpdesk ticket modal is the modal by which students can submit
+## new tickets to or review their ticket information
+#
+angular.module('doubtfire.helpdesk.modals.ticket-modal', [])
+
+.factory('HelpdeskTicketModal', ($modal) =>
+  HelpdeskTicketModal = {}
+
+  #
+  # Pass in the user, project
+  #
+  HelpdeskTicketModal.show = (ticket) =>
+    $modal.open
+      templateUrl: 'helpdesk/modals/ticket-modal/ticket-modal.tpl.html'
+      controller: 'HelpdeskTicketModal'
+      resolve:
+        ticket: => ticket
+
+  HelpdeskTicketModal
+)
+
+.controller('HelpdeskTicketModal', ($scope, $state, $rootScope, $modalInstance, HelpdeskTicket, ConfirmationModal, alert)
+  $scope.isNew = !ticket?
+  $scope.ticket = ticket || new HelpdeskTicket()
+
+  #
+  # Callback when a task definition has changed
+  #
+  $scope.taskDefSelected = (taskDef) =>
+    $scope.ticket.task_definition = taskDef
+
+  # Use project service to get projects
+  unless $scope.ticket.project?
+    projectService.getProjects (projects) =>
+      $scope.projects = projects
+      $scope.ticket.project = projects[0] if projects.length == 1
+      unless $scope.isNew
+        # Use the ticket information to find the correct project
+        projectService.findProject $scope.ticket.project_id, (p) =>
+          $scope.ticket.project = p
+
+  #
+  # Watch when a project is changed to update the selected unit
+  #
+  $scope.$watch 'ticket.project.project_id', (newId) =>
+    return unless newId?
+    $scope.taskDefSelected null if $scope.isNew # reset which task selected

```

```

49 +     unitService.getUnit $scope.ticket.project.unit_id, false, false, (response) ->
50 +       $scope.ticket.unit = response
51 +       # If ticket was provided, we need to look up the task def now from the
52 +       # unit loaded if it has one
53 +       if not $scope.isNew and $scope.ticket.task_definition_id?
54 +         taskDef = $scope.ticket.unit.taskDef($scope.ticket.task_definition_id)
55 +         $scope.taskDefSelected taskDef
56 +
57 +       #
58 +       # Sends a HTTP request to open a ticket
59 +       #
60 +       $scope.openTicket = ->
61 +         openTicketCallback = (error, success) ->
62 +           if success
63 +             $modalInstance.close(success)
64 +             $rootScope.$broadcast('CurrentOpenTicket', success)
65 +             alertService.add("success", "Ticket created successfully.", 2000)
66 +           if error
67 +             alertService.add("danger", "Error: #{error.data.error}", 6000)
68 +         HelpdeskTicket.submitTicket(
69 +           $scope.ticket.project.project_id,
70 +           $scope.ticket.description,
71 +           $scope.ticket.task_definition?.id,
72 +           openTicketCallback
73 +         )
74 +
75 +       #
76 +       # Sends a request to resolve the ticket
77 +       #
78 +       $scope.resolveTicket = ->
79 +         # TODO: Resolved by whom?
80 +         resolveTicketCallback = (error, success) ->
81 +           if success
82 +             $modalInstance.close(success)
83 +             $rootScope.$broadcast('CurrentOpenTicket', null)
84 +             alertService.add("success", "Ticket resolved!", 2000)
85 +           if error
86 +             alertService.add("danger", "Error: #{error.data.error}", 6000)
87 +         $scope.ticket.resolve(resolveTicketCallback)
88 +
89 +       #
90 +       # Sends a request to close the ticket
91 +       #
92 +       $scope.closeTicket = ->
93 +         closeTicketCallback = (error, success) ->
94 +           if success
95 +             $modalInstance.close(success)
96 +             $rootScope.$broadcast('CurrentOpenTicket', null)
97 +             alertService.add("success", "Ticket closed.", 2000)
98 +           if error
99 +             alertService.add("danger", "Error: #{error.data.error}", 6000)
100 +             ConfirmationModal.show 'Close Ticket', 'Are you sure you want to close the ticket without resolving the issue?', ->
101 +               $scope.ticket.close(closeTicketCallback)
102 +
103 +       #
104 +       # Goes to the selected task
105 +       #
106 +       $scope.goToSelectedTask = $modalInstance.dismiss
107 +     )

```

14 src/app/helpdesk/modals/ticket-modal/ticket-modal.scss

[View](#)

... ... @@ -0,0 +1,14 @@

```

1  +.ticket-modal {
2  +  .unit-selection .dropdown {
3  +    width: 100%;
4  +    button {
5  +      text-align: right;
6  +    }
7  +  }
8  +  label {
9  +    margin-right: 1ex;
10 +  }
11 +  .unit-selection .dropdown-menu > li > a {
12 +    padding: 1em;
13 +  }
14 +}

```

103 src/app/helpdesk/modals/ticket-modal/ticket-modal.tpl.html

[View](#)

```

...
@@ -0,0 +1,103 @@
1  +<div class="ticket-modal">
2  +  <form class="form-horizontal" role="form">
3  +    <div class="modal-header" ng-show="isNew">
4  +      <h3>Submit a Ticket</h3>
5  +      <p class="lead">
6  +        Having trouble with something? Get help from a tutor at the programming
7  +        helpdesk, located at ATC620.
8  +      </p>
9  +    </div><!--/new-ticket-header-->
10 +   <div class="modal-header" ng-hide="isNew">
11 +     <h3>Show Ticket</h3>
12 +   </div><!--/existing-ticket-header-->
13 +   <div class="modal-body">
14 +     <div class="form-group unit-selection">
15 +       <label class="col-sm-3 control-label">Unit</label>
16 +       <div class="col-sm-7">
17 +         <p class="form-control-static" ng-if="projects.length == 1 || !isNew">
18 +           <label class="label label-info small">{{ticket.project.unit_code}}</label>
19 +           <span class="unit-name">{{ticket.project.unit_name}}</span>
20 +         </p>
21 +         <div class="btn-group dropdown" ng-if="projects.length > 1 && isNew" dropdown is-open="status.isopen">
22 +           <button type="button" class="btn btn-default form-control dropdown-toggle" dropdown-toggle>
23 +             <span ng-show="ticket.project" class="pull-left">
24 +               <label class="label label-info small">{{ticket.project.unit_code}}</label>
25 +               <span class="unit-name">{{ticket.project.unit_name}}</span>
26 +             </span>
27 +             <span ng-hide="ticket.project" class="pull-left">
28 +               Select a Unit
29 +             </span>
30 +             <span class="caret"></span>
31 +           </button>
32 +           <ul class="dropdown-menu">
33 +             <li ng-repeat="project in projects">
34 +               <a ng-click="ticket.project = project">
35 +                 <label class="label label-info small">{{project.unit_code}}</label>
36 +                 <span class="unit-name">{{project.unit_name}}</span>
37 +               </a>
38 +             </li>
39 +           </ul>
40 +         </div>
41 +       </div>
42 +     </div><!--/unit-selection-->
43 +     <div class="form-group" ng-if="ticket.unit">
44 +       <label class="col-sm-3 control-label">Task</label>
45 +       <div class="col-sm-7" ng-if="isNew">

```

```

46 +     <task-definition-selector button-style="default" unit="ticket.unit" ng-model="ticket.task_definition" on-select="onTaskSelected">
47 +     <p class="help-block">
48 +       Select a task that you want help with, or leave blank if you have a
49 +       general question.
50 +     </p>
51 +   </div><!--/task-selector-if-new-->
52 +   <div class="col-sm-7" ng-if="!isNew">
53 +     <div ng-if="ticket.task_definition" class="selected-task-detail">
54 +       <label class="label label-info">{{ticket.task_definition.abbreviation}}</label>
55 +       {{ticket.task_definition.name}}
56 +     </div>
57 +     <div class="text-muted form-control-static" ng-hide="ticket.task_definition">
58 +       No task provided
59 +     </div>
60 +   </div>
61 + </div><!--/task-selection-->
62 + <div class="form-group">
63 +   <label class="col-sm-3 control-label">Description</label>
64 +   <div class="col-sm-7" ng-if="isNew">
65 +     <markdown-editor ng-model="ticket.description" placeholder="Enter description" height="100"></markdown-editor>
66 +     <p class="help-block">
67 +       Give the tutor some context to your issue by providing a description
68 +       of your question. {{description}}
69 +     </p>
70 +   </div>
71 +   <div class="col-sm-7" ng-if="!isNew">
72 +     <div ng-show="ticket.description" class="form-control-static" ng-bind-html="ticket.description | markdown"></div>
73 +     <p ng-hide="ticket.description" class="text-muted form-control-static">
74 +       No description provided
75 +     </p>
76 +   </div>
77 + </div><!--/description-->
78 + <div class="form-group" ng-if="!isNew">
79 +   <label class="col-sm-3 control-label">In Queue For</label>
80 +   <div class="col-sm-7 form-control-static">
81 +     {{ticket.lengthOfTimeOpen().humanize()}}
82 +     <small class="text-muted">
83 +       ({{ticket.created_at | date:'shortTime'}})
84 +     </small>
85 +   </div>
86 + </div>
87 + <div class="modal-footer text-right">
88 +   <button ng-click="$dismiss()" class="btn btn-primary">
89 +     Close
90 +   </button>
91 +   <button ng-disabled="ticket.project == null" ng-click="openTicket()" class="btn btn-success" ng-if="isNew">
92 +     <i class="fa fa-ticket"></i> <i class="fa fa-plus"></i> Submit Ticket
93 +   </button>
94 +   <button ng-click="closeTicket()" class="btn btn-danger" ng-if="!isNew">
95 +     <i class="fa fa-ticket"></i> <i class="fa fa-trash-o"></i> Close Ticket
96 +   </button>
97 +   <button ng-click="resolveTicket()" class="btn btn-success" ng-if="!isNew" if-role="Tutor Convenor Admin">
98 +     <i class="fa fa-ticket"></i> <i class="fa fa-check"></i> Resolve Ticket
99 +   </button>
100 + </div><!--/footer-->
101 + </form>
102 + </div>
103 +</div>

```

166

src/app/helpdesk/states/dashboard/dashboard.coffee

View



...

...

@@ -0,0 +1,166 @@

1
+#

```

2  ## The programming helpdesk dashboard
3  ##
4  +angular.module('doubtfire.helpdesk.states.dashboard', [])
5  +.config((headerServiceProvider) =>
6  +  helpdeskDashboardState =
7  +    url: "/helpdesk"
8  +    views:
9  +      main:
10 +        controller: "HelpdeskDashboardCtrl"
11 +        templateUrl: "helpdesk/states/dashboard/dashboard.tpl.html"
12 +
13 +      data:
14 +        pageTitle: "_Programming Helpdesk_"
15 +        roleWhitelist: ['Student', 'Tutor', 'Convenor', 'Admin']
16 +  headerServiceProvider.state 'helpdesk', helpdeskDashboardState
17 +
18 +.controller("HelpdeskDashboardCtrl", ($scope, $timeout, $rootScope, $interval, $state, HelpdeskStats, HelpdeskTicket, HelpdeskStaff)
19 +  # Internal poll interval
20 +  pollInterval = null
21 +  # How long the duration is between each poll, in seconds
22 +  intervalDuration = 30
23 +
24 +  #
25 +  # Returns true if the stats are already being polled
26 +  #
27 +  isPolling = -> pollInterval isnt null
28 +
29 +  # Keep track of new stats|staff|tickets in these variable
30 +  $scope.data =
31 +    tickets: []
32 +    tutorsWorking: []
33 +
34 +  # This function is called when tickets have been updated
35 +  #
36 +  ticketsUpdated = (error, tickets) ->
37 +    # Only update what we need to
38 +    resolvedTickets = _.differenceBy($scope.data.tickets, tickets, 'id')
39 +    newTickets = _.differenceBy(tickets, $scope.data.tickets, 'id')
40 +    # Remove resolved tickets + merge new tickets
41 +    $scope.data.tickets =
42 +      _.chain($scope.data.tickets)
43 +        .filter((t) -> !_.find(resolvedTickets, { id: t.id })?)
44 +        .concat(newTickets)
45 +        .map((ticket, idx) ->
46 +          # Add ticket's position
47 +          ticket.position = idx + 1
48 +          ticket
49 +        )
50 +        .value()
51 +
52 +  #
53 +  # This function is called when stats have been updated
54 +  #
55 +  statsUpdated = (error, stats) ->
56 +    statusColours = {'Quiet': 'primary', 'Available': 'warning', 'BUSY': 'danger'}
57 +    statusEmojis = {'Quiet': '😊', 'Available': '😊', 'BUSY': '😱'}
58 +    statsTimes = _.keys(stats)
59 +    lastRecord = stats[_.last(statsTimes)]
60 +    avgWaitTime = $scope.avgWaitTime = lastRecord.average_wait_time_in_mins
61 +    numUnresolved = $scope.numUnresolved = lastRecord.number_of_unresolved_tickets
62 +    # TODO: Work out the right values
63 +    $scope.averageWaitTimeStatus =
64 +      if avgWaitTime < 6

```

```

65      'Quiet'
66  else if 6 <= avgWaitTime < 9
67      'Available'
68  else if avgWaitTime > 9
69      'BUSY'
70  numberUnresolvedStatus =
71  if numUnresolved < 6
72      'Quiet'
73  else if 6 <= numUnresolved < 9
74      'Available'
75  else if numUnresolved > 9
76      'BUSY'
77  $scope.averageWaitTimeEmoji = statusEmojis[$scope.averageWaitTimeStatus]
78  $scope.averageWaitTimeColor = statusColours[$scope.averageWaitTimeStatus]
79  $scope.numberUnresolvedColor = statusColours[numberUnresolvedStatus]
80  # Angular nvd3 accepts milliseconds not seconds for unix time
81  $scope.graphData = stats
82 +
83 +
84  # This function is called when staff have been updated
85 +
86  tutorsUpdated = (error, sessions) ->
87  # Same concept as ticketsUpdated
88  sessionsFinished = _.differenceBy($scope.data.tutorsWorking, sessions, 'id')
89  sessionsStarted = _.differenceBy(sessions, $scope.data.tutorsWorking, 'id')
90  $scope.data.tutorsWorking =
91  _.chain($scope.data.tutorsWorking)
92  .filter((s) -> !_.find(sessionsFinished, { id: s.id })?)
93  .concat(sessionsStarted)
94  .map((s) ->
95    finishIn = s.timeUntilFinish()
96    s.minuteDisplay = Math.round(finishIn.minutes() + (finishIn.seconds() * 0.01))
97    s.hourDisplay = finishIn.hours()
98    if s.minuteDisplay == 60
99      s.minuteDisplay = 0
100     s.hourDisplay += 1
101    s.showHour = s.hourDisplay > 0
102    s
103  )
104  .value()
105  $scope.helpdeskClosed = $scope.data.tutorsWorking.length is 0
106  $scope.helpdeskOpen = not $scope.helpdeskClosed
107 +
108 +
109  # If clock on event, refresh immediately
110 +
111  $rootScope.$on 'CurrentWorkingSession', (event, session) ->
112  $scope.pollNow()
113 +
114 +
115  # Begins polling for stats
116 +
117  startPolling = (interval = intervalDuration) ->
118  return if isPolling()
119  pollForStats = ->
120  # We know if it's the first poll if we are not polling yet!
121  isFirstInterval = not isPolling()
122  # First poll we want to get the last hour's worth
123  subtractValue = if isFirstInterval then 1 * 60 * 60 else interval
124  useInterval = if isFirstInterval then null else interval
125  from = moment().subtract(subtractValue, 'seconds').format()
126  to = moment().format()
127  HelpdeskStats.get from, to, useInterval, (error, response) ->

```

```

128 +     $scope.lastUpdated = moment()
129 +     statsUpdated(error, response)
130 +
131 +     pollForTickets = ->
132 +       HelpdeskTicket.getUnresolvedTickets null, (error, response) ->
133 +         $scope.lastUpdated = moment()
134 +         ticketsUpdated(error, response)
135 +     pollForTutors = ->
136 +       HelpdeskSession.tutorsWorkingNow (error, response) ->
137 +         $scope.lastUpdated = moment()
138 +         tutorsUpdated(error, response)
139 +     pollFunction = ->
140 +       pollForTickets()
141 +       pollForTutors()
142 +       pollForStats()
143 +     $scope.pollNow = ->
144 +       pollFunction()
145 +     # Call poll at least once to start now
146 +     $scope.pollNow()
147 +
148 +
149 +   # Stops helpdesk statistics from polling
150 +
151 +   stopPolling = ->
152 +     return unless isPolling()
153 +     $interval.cancel(pollInterval)
154 +     pollInterval = null
155 +
156 +
157 +
158 +   # Opens the session modal
159 +
160 +   $scope.openHelpdeskSessionModal = HelpdeskSessionModal.show
161 +
162 +   # When we load, start polling at intervalDuration seconds
163 +   startPolling(intervalDuration)
164 +   # When we unload, stop polling
165 +   $scope.$on '$destroy', stopPolling
166 + )

```

105 src/app/helpdesk/states/dashboard/dashboard.scss

[View](#)

```

...
...
@@ -0,0 +1,105 @@
1 // This mixin is used to hide on mobile devices specific rows
2 +@ mixin hide-on-mobile {
3 +  @extend .hidden-xs;
4 +  @extend .hidden-sm;
5 +
6 +
7 +#helpdesk-dashboard {
8 +  // Fix layout for medium
9 +  @media (min-width: $screen-md) {
10 +    height: calc(100vh - #{$navbar-height});
11 +    min-height: 600px;
12 +    display: flex;
13 +    flex-direction: column;
14 +    .row {
15 +      display: flex;
16 +    }
17 +    .status.row {
18 +      flex: 1;
19 +    }
20 +    .graph.row {

```

```
21 +     flex: 2;
22 +
23 +     .ticket.row {
24 +       flex: 1.5;
25 +     }
26 +
27 +
28 + // Hide all rows on mobile except the tickets row
29 + .row:not([class^="ticket"]) {
30 +   @include hide-on-mobile;
31 + }
32 +
33 + // Give rows some space between eachother
34 + .row {
35 +   margin: 15px auto 0 auto;
36 + }
37 +
38 + // Help Header
39 + h3.help {
40 +   @extend .text-muted;
41 +   margin-top: 0;
42 +   text-transform: uppercase;
43 +   font-weight: bold;
44 +   font-size: 1.2em;
45 + }
46 +
47 + // Status Row
48 + .status.row {
49 +   font-size: 2em;
50 +   text-align: center;
51 +   .number {
52 +     display: block;
53 +     font-size: 4em;
54 +     line-height: 1;
55 +   }
56 +
57 + // Tutor panel
58 + .tutors-working {
59 +   align-self: center;
60 +   .list-group {
61 +     max-height: 300px;
62 +     overflow-y: scroll;
63 +   }
64 +   .list-group-item-heading {
65 +     font-size: 1.5em;
66 +     .name {
67 +       font-weight: bold;
68 +     }
69 +   }
70 +   .list-group-item-text {
71 +     margin: 0.7em auto 1em auto;
72 +     font-size: 1.2em;
73 +     display: flex;
74 +     flex-wrap: wrap;
75 +     .label:not(:last-of-type) {
76 +       margin-right: 1ex;
77 +       margin-bottom: 1ex;
78 +     }
79 +   }
80 + // Ticket Row
81 + .ticket.row {
82 +   justify-content: center;
83 + }
```

```

84 + // Helpdesk Closed
85 + .helpdesk-closed {
86 +   height: 100%;
87 +   margin-top: 2em;
88 +   display: flex;
89 +   align-items: center;
90 +   text-align: center;
91 +   font-size: 1.5em;
92 +   .message {
93 +     margin: 0 auto;
94 +     button {
95 +       margin-top: 1em;
96 +     }
97 +     .title {
98 +       font-weight: bold;
99 +       display: block;
100 +      margin-top: 1em;
101 +      font-size: 1.5em;
102 +    }
103 +  }
104 + }
105 +}

```

74 src/app/helpdesk/states/dashboard/dashboard.tpl.html

[View](#)

```

...
...
@@ -0,0 +1,74 @@
1 +<div id="helpdesk-dashboard" class="clearfix">
2 +  <div class="helpdesk-closed">
3 +    <div class="message" ng-if="helpdeskClosed">
4 +      <i class='fa fa-hourglass-end fa-4x'></i>
5 +      <span class="title"> Helpdesk Closed </span>
6 +      No tutors are working right now<br>
7 +      <button ng-click="openHelpdeskSessionModal()" if-role="Tutor Convenor Admin" class="btn btn-lg btn-success">
8 +        <i class="fa fa-clock-o"></i> <i class="fa fa-sign-in"></i> Clock On
9 +      </button>
10 +    </div>
11 +    <div class="message" ng-if="helpdeskClosed == null && helpdeskOpen == null">
12 +      <i class="fa fa-spinner fa-pulse fa-4x"></i>
13 +    </div>
14 +  </div><!--/helpdesk-closed-->
15 +  <div ng-if="helpdeskOpen">
16 +    <div class="status row">
17 +      <div class="people-in-queue col-sm-4 text-{{numberUnresolvedColor}}">
18 +        <span class="number">
19 +          {{numUnresolved}}
20 +        </span>
21 +        people in queue
22 +      </div><!--/people-in-queue-->
23 +      <div class="emoji-face col-sm-4">
24 +        <span class="number">{{averageWaitTimeEmoji}}</span> {{averageWaitTimeStatus}}
25 +      </div><!--/emoji-face-->
26 +      <div class="average-wait-time col-sm-4 text-{{averageWaitTimeColor}}">
27 +        <span class="number">
28 +          {{avgWaitTime}}m
29 +        </span>
30 +        average wait time
31 +      </div><!--/average-wait-time-->
32 +    </div><!--/status-row-->
33 +    <div class="graph row">
34 +      <div class="tutors-working col-sm-4">
35 +        <h3 class="help">
36 +          {{ tutorCount = data.tutorsWorking.length }}
37 +          tutor{{tutorCount > 1 ? 's' : ''}} working right now

```

```

38 +     </h3>
39 +     <div class="list-group">
40 +       <div ng-repeat="session in data.tutorsWorking" class="list-group-item">
41 +         <div class="list-group-item-heading">
42 +           <span class="name">
43 +             {{finishIn = session.timeUntilFinish(); session.user.name}}
44 +           </span><!--tutor-name-->
45 +           <div class="finish-in pull-right text-muted pointer" tooltip="Finishes at {{session.clock_off_time | date}}">
46 +             <span ng-if="session.showHour">
47 +               {{session.hourDisplay}}h
48 +             </span>
49 +             {{session.minuteDisplay}}m
50 +           </div><!--finish-in-->
51 +         </div>
52 +         <p class="list-group-item-text">
53 +           <label ng-repeat="unit in session.user.units_taught" class="label label-info pointer" tooltip="{{unit.name}}: {{unit.code}}">
54 +             {{unit.code}}
55 +           </label>
56 +           </p><!--unit-labels-->
57 +         </div>
58 +       </div><!--tutors-working-list-->
59 +     </div><!--tutors-working-->
60 +     <div class="graph col-sm-8">
61 +       <h3 class="help">
62 +         Tickets unresolved and wait times today
63 +       </h3>
64 +       <helpdesk-dashboard-timeline data="graphData" height="400">
65 +       </helpdesk-dashboard-timeline>
66 +     </div>
67 +   </div><!--graph-row-->
68 +   <div class="ticket_row">
69 +     <div class="ticket-wrapper col-sm-3" ng-repeat="ticket in data.tickets">
70 +       <helpdesk-ticket ticket="ticket"></helpdesk-ticket>
71 +     </div>
72 +   </div><!--ticket-row-->
73 + </div><!--helpdesk-open-->
74 +</div>

```

3 src/app/helpdesk/states/states.coffee

[View](#)

```

...
...
@@ -0,0 +1,3 @@
1 +angular.module('doubtfire.helpdesk.states', [
2 +  'doubtfire.helpdesk.states.dashboard'
3 +])

```

2 src/app/tasks/task-definition-selector/task-definition-selector.coffee

[View](#)

```

@@ -11,6 +11,8 @@ angular.module('doubtfire.tasks.task-definition-selector', [])
11 11   scope:
12 12     # Unit required
13 13     unit: "="
14 14     # Model
15 15     selectedDefinition: '=ngModel'
16 16     # What to do when definition is changed
17 17     onSelectDefinition: "="
18 18     # Provide a btn-style to force the colour to change`
```

96 src/app/visualisations/helpdesk-dashboard-timeline.coffee

[View](#)

```

...
...
@@ -0,0 +1,96 @@
1 +angular.module('doubtfire.visualisations.helpdesk-dashboard-timeline', [])
2 +.directive 'helpdeskDashboardTimeline', >
```

```

3 + replace: true
4 + restrict: 'E'
5 + templateUrl: 'visualisations/visualisation.tpl.html'
6 + scope:
7 +   rawData: '=data'
8 +   height: '=?'
9 + controller: ($scope, $timeout, taskService, projectService, Visualisation) ->
10 +   updateData = (rawData) ->
11 +     # Function to fix time as nvd3 needs a unix time in ms, not s
12 +     correctTime = (t) -> t * 1000
13 +     $scope.times = _.chain(rawData).keys().map(correctTime).value()
14 +     $scope.data = [
15 +       {
16 +         key: 'Average Wait Time',
17 +         bar: false,
18 +         values: _.map rawData, (data, time) ->
19 +           {
20 +             x: correctTime(time),
21 +             y: data.average_wait_time_in_mins
22 +           }
23 +         },
24 +         {
25 +           key: 'People In Queue',
26 +           bar: true,
27 +           values: _.map rawData, (data, time) ->
28 +             {
29 +               x: correctTime(time),
30 +               y: data.number_of_unresolved_tickets
31 +             }
32 +         }
33 +       ]
34 +     # Update forceY to be the max num
35 +     barMax =
36 +       _.chain(rawData)
37 +         .map((data) -> data.number_of_unresolved_tickets)
38 +         .max()
39 +     barMax = if barMax > 10 then barMax else 10
40 +     lineMax =
41 +       _.chain(rawData)
42 +         .map((data) -> data.average_wait_time_in_mins)
43 +         .max()
44 +     lineMax = if lineMax > 10 then lineMax else 10
45 +     $scope.options.chart.bars.forceY =
46 +     $scope.options.chart.bars2.forceY =
47 +       [0, barMax]
48 +     $scope.options.chart.lines.forceY =
49 +     $scope.options.chart.lines2.forceY =
50 +       [0, lineMax]
51 +     $timeout ->
52 +       $scope.api.refresh() if $scope.api?.refresh?
53 +
54 +     $scope.$watch (-> _.keys($scope.rawData)[0]), ->
55 +       return unless $scope.rawData?
56 +       updateData $scope.rawData
57 +
58 +     [$scope.options, $scope.config] = Visualisation 'linePlusBarChart', 'Helpdesk Dashboard Timeline', {
59 +       showLabels: no
60 +       margin:
61 +         top: 30,
62 +         right: 30,
63 +         bottom: 50,
64 +         left: 70
65 +       height: $scope.height

```

```

66 +   bars:
67 +     forceY: [0]
68 +   bars2:
69 +     forceY: [0]
70 +   lines:
71 +     forceY: [0]
72 +   lines2:
73 +     forceY: [0]
74 +   color: [
75 +     'darkred',
76 +     '#2ca02c'
77 +   ]
78 +   x: (d, i) -> i
79 +   xAxis:
80 +     axisLabel: 'Time'
81 +     tickFormat: (d) -> moment($scope.times[d]).format('hh:mm a')
82 +   x2Axis:
83 +     showMaxMin: true
84 +     tickFormat: (d) -> moment($scope.times[d]).format('hh:mm a')
85 +   y1Axis:
86 +     axisLabel: 'people waiting'
87 +     tickFormat: (d) -> d
88 +     axisLabelDistance: -12
89 +   y2Axis:
90 +     axisLabel: null
91 +     tickFormat: (d) -> "#{d}m"
92 +   y3Axis:
93 +     tickFormat: (d) -> 'l'
94 +   y4Axis:
95 +     tickFormat: (d) -> 'x'
96 + }, {}

```

1 src/app/visualisations/visualisations.coffee

[View](#)

```

diff --git a/src/app/visualisations/visualisations.coffee b/src/app/visualisations/visualisations.coffee
@@ -9,6 +9,7 @@ angular.module('doubtfire.visualisations', [
 9   'doubtfire.visualisations.task-completion-box-plot'
10  'doubtfire.visualisations.achievement-box-plot'
11  'doubtfire.visualisations.achievement-custom-bar-chart'
12 + 'doubtfire.visualisations.helpdesk-dashboard-timeline'
13 ])
14
15 .factory('Visualisation', ($interval, analyticsService) ->

```

7 src/styles/common/alerts.scss

[View](#)

```

diff --git a/src/styles/common/alerts.scss b/src/styles/common/alerts.scss
@@ -49,3 +49,10 @@
49   .alert.alert-warning i.fa:before { content: $fa-var-exclamation-triangle; }
50   .alert.alert-success i.fa:before { content: $fa-var-check-circle; }
51 }
52 +
53 // Push down the alert past the header if it has content
54 +@media (min-width: $screen-sm) {
55 +  div[ui-view="header"]:not(:empty) ~ .alert-top {
56 +    top: 66px;
57 +  }
58 +}

```

8 src/styles/common/overrides/modal-overrides.scss

[View](#)

```

diff --git a/src/styles/common/overrides/modal-overrides.scss b/src/styles/common/overrides/modal-overrides.scss
@@ -0,0 +1,8 @@
1  // Mobile responsive fixes
2 +@media (max-width: $screen-md) {

```

```
3 + .modal-footer button {
4 +   width: 100%;
5 +   margin-bottom: 1em !important;
6 +   margin-left: 0 !important;
7 +
8 +}
```



Chapter 13

Meeting Minutes

Contents

1 Meeting 9 Aug 11:30am	4
1.1 Present	4
1.2 Agenda Items	4
1.3 Discussion, Decisions and Agreements	4
1.3.1 RE: 1. Debrief of changes over holiday	4
1.3.2 RE: 2. Possible alterations to design	5
1.3.3 RE: 3. Discuss what has been completed	7
1.3.4 RE: 4. Discuss what needs to be done next	8
1.4 Agenda Items For Next Meeting	9
1.4.1 Follow Up Actions	9
2 Meeting 9 Aug at 1:00pm	9
2.1 Present	9
2.2 Agenda Items	9
2.3 Discussion, Decisions and Agreements	9
2.3.1 RE: 1. Feedback from last semester	9
2.3.2 RE: 2. Required Documentation	10
2.3.3 RE: 3. Regular meetings	10
2.4 Agenda Items For Next Meeting	10
2.4.1 Follow Up Actions	10
3 Meeting 16 Aug 12:00pm	10
3.1 Present	10
3.2 Agenda Items	11
3.3 Discussion, Decisions and Agreements	11
3.3.1 RE: 1. API additions	11
3.3.2 RE: 2. Progress update on UI	12
3.3.3 RE: 3. Wireframes and Ticket Directive Design	12
3.3.4 Additional Items	13
3.4 Agenda Items For Next Meeting	13
3.4.1 Follow Up Actions	13

4 Meeting 23 Aug 11:30am	14
4.1 Present	14
4.2 Agenda Items	14
4.3 Discussion, Decisions and Agreements	14
4.3.1 RE: 1. Update on Ticketing System implementation	14
4.3.2 RE: 2. Dashboard UI Prototype	14
4.3.3 RE: 3. Usability Testing	15
4.3.4 RE: 4. Timeline for the coming weeks	15
4.4 Agenda Items For Next Meeting	15
4.4.1 Follow Up Actions	17
5 Meeting 30 Aug 12:00pm	17
5.1 Present	17
5.2 Agenda Items	17
5.3 Discussion, Decisions and Agreements	17
5.3.1 RE: 1. Review Sprint Plan	17
5.3.2 RE: 2. Review Changes to Codebase	17
5.3.3 Additional Items	18
5.4 Agenda Items For Next Meeting	18
5.4.1 Follow Up Actions	18
6 Meeting 6 Sept 12:00pm	18
6.1 Present	18
6.2 Agenda Items	18
6.3 Discussion, Decisions and Agreements	19
6.3.1 RE: 1. Progress on dashboard framework	19
6.3.2 RE: 2. Usability evaluation test planning	19
6.3.3 Additional Items	20
6.4 Agenda Items For Next Meeting	20
6.4.1 Follow Up Actions	21
7 Meeting 20 Sept 10:15am	21
7.1 Present	21
7.2 Agenda Items	21

7.3	Discussion, Decisions and Agreements	21
7.3.1	RE: 1. Progress update	21
7.3.2	Additional Items	21
7.4	Agenda Items For Next Meeting	22
7.4.1	Follow Up Actions	22
8	Meeting 4 October 11:15am	22
8.1	Present	22
8.2	Agenda Items	22
8.3	Discussion, Decisions and Agreements	22
8.4	RE: 1. Planning For Video	22
8.5	RE: 2. Compiling portfolio	24
8.6	RE: 3. Finish usability document	24
8.7	RE: 4. Who did What document	24
8.7.1	Additional Items	24
8.8	Agenda Items For Next Meeting	24
8.8.1	Follow Up Actions	25

1 Meeting 9 Aug 11:30am

Location: ATC Project Rooms

1.1 Present

- Alex Cummaudo
- Jake Renzella
- Lachlan West
- Reuben Wilson

1.2 Agenda Items

1. Debrief of changes over holiday
2. Possible alterations to design
3. Discuss what has been completed
4. Discuss what needs to be done next

1.3 Discussion, Decisions and Agreements

1.3.1 RE: 1. Debrief of changes over holiday

- During the break Andrew Cain, Alex, Jake and Lachlan discussed breaking down the project into three phases:
 1. First iteration is an asynchronous system with which tutors can use the ticketing system from the helpdesk terminal
 2. Second iteration is to make the web front-end more mobile-friendly so that staff can use the ticketing system directly from their smartphones. Possible to get the first two done together if working in a mobile-first development scheme.
 3. Third iteration is to look into developing smartphone applications with an synchronous, event-driven API
- Andrew Cain discouraged iOS/Android apps as it will be considered hard to maintain with the app cycle
- Keep the design integrated into the existing web system

- Andrew can then maintain and upkeep this
- Developing through the Apple App Store/Google Play Store is a long process and will require future bug-releases which will be hard to push to all users
- If we maintain the single system it will be easier to develop and push changes once our team has finished the project without much overhead for other developers
- Dropping the synchronous event-driven micro service until phase 3

1.3.2 RE: 2. Possible alterations to design

- We probably don't need a queue at all and thus don't have a need for a separate server. Instead we sort all tickets by a date/time field.
- Refer to the Trello board, specifically these two cards:
 1. Rethink Backend Implementation¹
 2. Public Doubtfire Ticketing²
- We need to add a Helpdesk dropdown menu. Let's use fa fa-support for the icon:

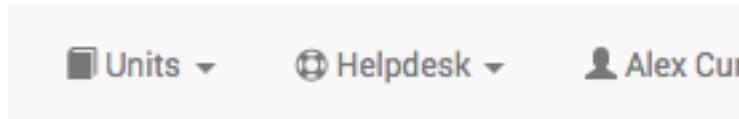


Figure 1: Helpdesk Dropdown

- If I am a student in a unit, I should see **Create a Ticket if I don't currently have a ticket open**. This would then open the Create a Ticket Modal.
- If I teach a unit, I should see **Clock On** or **Clock Off** (depending on if I am already clocked on or not). This would then open the Clock On/Clock Off modal:
- I should always be able to view the helpdesk status. This would take me to the Helpdesk Dashboard.
- Modals used to create new tickets and clock on/clock off staff members

¹See <https://trello.com/c/vKMN6hvN/47-rethink-doubtfire-backend-implementation>

²See <https://trello.com/c/qZJ2j4zQ/65-public-doubtfire-ticketing>

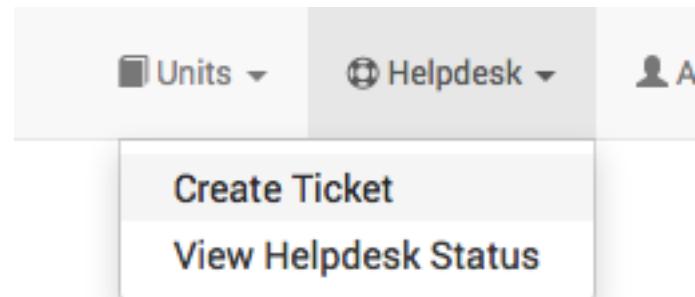


Figure 2: Create a ticket

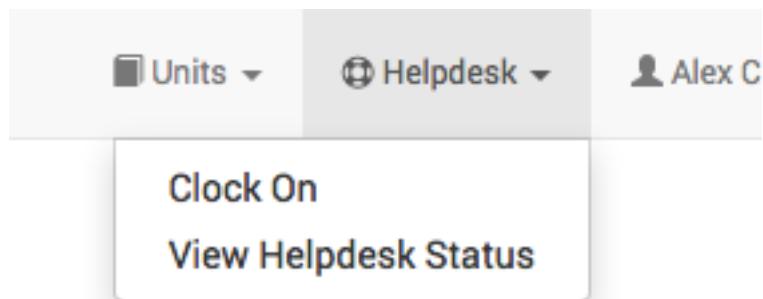


Figure 3: Clock on

- Only issue becomes authentication vulnerabilities:
- If we are using the helpdesk terminal in phase one, one staff member needs to login. E.g., Fred logs in so that the dashboard is live. Then when Jane resolves a ticket, she is resolving it under Fred's account. Vulnerability for abuse.
- Andrew has discussed the idea of each staff member having a unique PIN to authorise helpdesk transactions
- But for initial phases this isn't really necessary as each helpdesk user is most likely genuine

1.3.3 RE: 3. Discuss what has been completed

- Jake and Alex have been coming in most of the holidays to work on the API
- New models include `HelpdeskTicket` and `HelpdeskSession`.
- `HelpdeskTicket` refers to a single helpdesk ticket that a student creates. It consists of:
 1. A required project (i.e., unit the student needs help with)
 2. An optional task definition (optional because they might need help with just setting up their computers, a common question asked in the initial weeks)
 3. A description of what they need help with
- `HelpdeskSession` refers to the session by which a staff member works at the helpdesk. It consists of:
 1. The time the staff member clocks onto the helpdesk (always the current time of creation)
 2. The *estimated* clock off time (estimated in case a staff member needs to leave immediately)
 3. The id of the user who clocks on and clocks off
- Endpoints for these can be found in swagger on the branch `new/helpdesk-ticketing`³, which include:
 - **GET /api/helpdesk/tickets** Gets all helpdesk tickets
 - **POST /api/helpdesk/tickets** Add a new helpdesk ticket
 - **GET /api/helpdesk/tickets/{id}** Gets helpdesk ticket with an id

³See <https://github.com/final-year-project/doubtfire-api/tree/new/helpdesk-ticketing>

- **PUT /api/helpdesk/tickets/{id}** Updates helpdesk ticket with an id
- **GET /api/helpdesk/sessions** Get helpdesk sessions
- **POST /api/helpdesk/sessions** Begin a new session at the helpdesk as current user
- **DELETE /api/helpdesk/sessions/{id}** Prematurely clock off an existing helpdesk session
- **GET /api/helpdesk/sessions/tutors** Get a list of all currently tutors working at the helpdesk
- **GET /api/helpdesk/stats** Gets statistics about the helpdesk for the duration specified
- Alex went over each of the above

1.3.4 RE: 4. Discuss what needs to be done next

- Unit testing for API is still needed. Discussed that Jake would probably be best as he knows the most on the unit testing framework.
- Web front end, with specific changes:
 1. Helpdesk Dropdown Menu in header bar
 2. Create a ticket modal
 3. Clock on/clock off modal
 4. Helpdesk Dashboard
- Usability testing
- There should be some form of usability testing with mockups on what the helpdesk dashboard should look like
- The other web front-end features are basic form entry and shouldn't require much usability testing
- Will need to ensure that the dashboard is clear and easy to read and understand
- There will need to be some investigation into graphically representing statistics
- Alex will come up with some cards for these tasks, to be worked on by Reuben and Lachlan
- Jake and Alex will maintain and improve the API as the semester goes on
- Follow up with Andrew Cain regarding PIN entry

1.4 Agenda Items For Next Meeting

1. Progress update from Lachlan and Reuben on work they have carried out

1.4.1 Follow Up Actions

1. **Alex - ASAP** - Create Trello Cards
 2. **Alex - ASAP** - Organise another weekly meeting for next week
-

2 Meeting 9 Aug at 1:00pm

Location: Graham's office

2.1 Present

- Alex Cummaudo
- Jake Renzella
- Lachlan West
- Reuben Wilson
- Graham Farrell

2.2 Agenda Items

1. Feedback from last semester
2. Required documentation this semester
3. Regular meetings

2.3 Discussion, Decisions and Agreements

2.3.1 RE: 1. Feedback from last semester

- Touch base on where we are at from last semester
- Graham was happy with our work
- Think we did well under the circumstances

2.3.2 RE: 2. Required Documentation

- Continue on project plan from last semester
- Make changes as we see fit
- **Graham needs something tangible**
- Do a lot for the usability testing!
- We will know more come Friday's lecture

2.3.3 RE: 3. Regular meetings

- Graham particularly busy this semester
- Best if we arrange meetings on Tuesday around midday
- Touch base every couple of weeks
- Alex to send him progress updates in dot points weekly

2.4 Agenda Items For Next Meeting

1. Progress updates

2.4.1 Follow Up Actions

1. [Alex] - [Next Wednesday] - Send Graham a progress update after about a week or so
 2. [Alex] - [ASAP] - Prepare wiki for Semester 2
-

3 Meeting 16 Aug 12:00pm

Location: ATC Project Rooms

3.1 Present

- Alex Cummaudo
- Jake Renzella
- Lachlan West
- Reuben Wilson

3.2 Agenda Items

1. API additions
2. Progress Update on UI
3. Wireframes and Ticket Directive Design

3.3 Discussion, Decisions and Agreements

3.3.1 RE: 1. API additions

- Reuben has implemented ticket API endpoint to return all tickets by one student.
This endpoint is ‘GET /helpdesk/user/:user_id/ticket’
- The intended use for this comes from Andrew Cain: **Students shouldn’t be able to post a new ticket to the helpdesk if they have already got a ticket open.**
- Reuben has submitted a PR for code review on new API endpoint into `new/helpdesk-ticketing`.
- Alex will code review the PR
- New functionality includes an added a shallow serializer for getting tickets which Andrew Cain suggested.
- Alex suggested that the response back from the database should sort the tickets based on ascending time order
- Reuben will look into this
- Lachlan has asked how to modify the serialiser which Reuben will help based on his recent learning
- Still missing are unit tests
- Jake can assist with writing these unit tests
- Alex suggested that we prevent POSTing a new ticket if a student has already got a ticket open. This needs to be added.
- Jake suggested wrapping the model functionality into the model file, not the API file
- This way we can reuse the same functionality in both the POST and GET endpoints
- Notes for code review:
- Remove new endpoint and use `GET /helpdesk/ticket` with `filter` query parameter

- Add `user_id` parameter to this endpoint - achieves the same functionality
- Don't have model code in API code
- Usage would be `GET /helpdesk/ticket?user_id=1&filter=unresolved`

3.3.2 RE: 2. Progress update on UI

- Modal is coming along nicely
- Caching information may be a cause for concern
- If a user refreshes the browser, do they need to send a new API request?
- Instead, couldn't we just cache what we know
- Either wrap this in the API file or cache with angular cookies
 - Use unit-service as model for caching tickets - don't write them in services however as this model is deprecated
- Discuss this with Andrew first
- Functionality is needed over 'niceness'!
- Lachlan will finish the modal first
- Need to begin working on `Tickets Opened` modal
- Show this in the dropdown menu when a ticket is opened
- When tickets are open, we should eliminate units from dropdown who have open tickets under the `Create Ticket` modal

3.3.3 RE: 3. Wireframes and Ticket Directive Design

- Create a `<helpdesk-ticket>` directive
- Takes in the `ticket-id="123"`
- The directive can make the API request, showing a spinner during the request
- We can reuse the ticket on the dashboard as well as in the new "Tickets Opened"
- Initially, we can use a BS `.well` class to show ticket information
- Ticket to show brief information under expandable accordion, such as:
- Student name
- Unit (and optionally the task)
- Student target grade
- Time ticket was opened
- How long the ticket has been opened for

- Can think of more later...
- Use `if-role` attribute directive to check the role for viewing tickets
- When `if-role='Tutor Convenor Admin'`, we can show the `Resolve` button
- When staff click resolve, on ticket ask who is resolving ticket from active tutors at helpdesk
- Don't worry about CSS too much for now.

3.3.4 Additional Items

- Alex and Jake will make wireframes for the dashboard design
- How does a tutor clock on and off from HD?
- On Trello, there is a card with details

3.4 Agenda Items For Next Meeting

1. Progress review
2. Organise another meeting with Andrew and Graham

3.4.1 Follow Up Actions

1. **Reuben - ASAP** - Sort the tickets by date created in endpoint
 2. **Reuben, Lachlan - ASAP** - Get together and show Lachlan how to modify serialiser
 3. **Jake, Reuben - Once PR has gone through** - Write unit tests for the endpoint
 4. **Alex or Reuben - After code review** - Write a way that prevents students from POSTing a ticket if they already have a ticket open
 5. **Reuben or Lachlan - ASAP** - Begin working on `Tickets Opened` modal
 6. **Reuben or Lachlan - ASAP** - Remove units from dropdown if it has a ticket opened
 7. **Reuben and Lachlan - Start soon** - Create a `<helpdesk-ticket>` directive (see minutes)
 8. **Alex - 16 Aug** - Code review
 9. **Reuben, Lachlan - Planning later soon** - Sessions modal
-

4 Meeting 23 Aug 11:30am

Location: ATC Project Rooms

4.1 Present

- Alex Cummaudo
- Jake Renzella
- Lachlan West
- Reuben Wilson

4.2 Agenda Items

1. Update on Ticketing System implementation
2. Dashboard UI Prototype
3. Usability Testing
4. Timeline for the coming weeks

4.3 Discussion, Decisions and Agreements

4.3.1 RE: 1. Update on Ticketing System implementation

- Velocity on the UI has stalled a bit
- Reuben and Lachy having bit of trouble with the foundations
- Alex suggested to take over development for a week
 - Build the bare-bones foundation so that Lachy and Reuben can build upon it
 - Aim to have the modals done, and have the states laid out
 - Changes would be to enhance quality, make it more functional
 - Lachy and Reuben will build upon the UI once they have more feedback following usability testing

4.3.2 RE: 2. Dashboard UI Prototype

- Jake and Alex will design the UI for the dashboard
 - Initially just a wireframe will do

- Eventually a small prototype would be nice (designed in Sketch) – but not essential
- The Sketch design would only assist development, but communication between the designers and the developers can always resolve this

4.3.3 RE: 3. Usability Testing

- Jake, Reuben, Lachlan and Alex will go to the help desk
 - Ask end-users for their opinions to create wireframes
 - Can also ask for additional ideas
 - What would the dashboard best look like for tutors? Students?
 - Think of a survey for opinions
- Actual testing can be done by recording people using the ticketing system
- Just use screen-capture software

4.3.4 RE: 4. Timeline for the coming weeks

- Create a timeline of the following weeks
- Alex made a mock on the whiteboard
- Weeks:
 - 4-5 development
 - 6-7 usability
 - 8-10: code changes based on usability feedback
 - 11-12: presentation & finalise everything
- Refer to whiteboard:
- Alex will make this legible and share next meeting

4.4 Agenda Items For Next Meeting

1. Sprint Plan Review
2. Review Alex's Changes

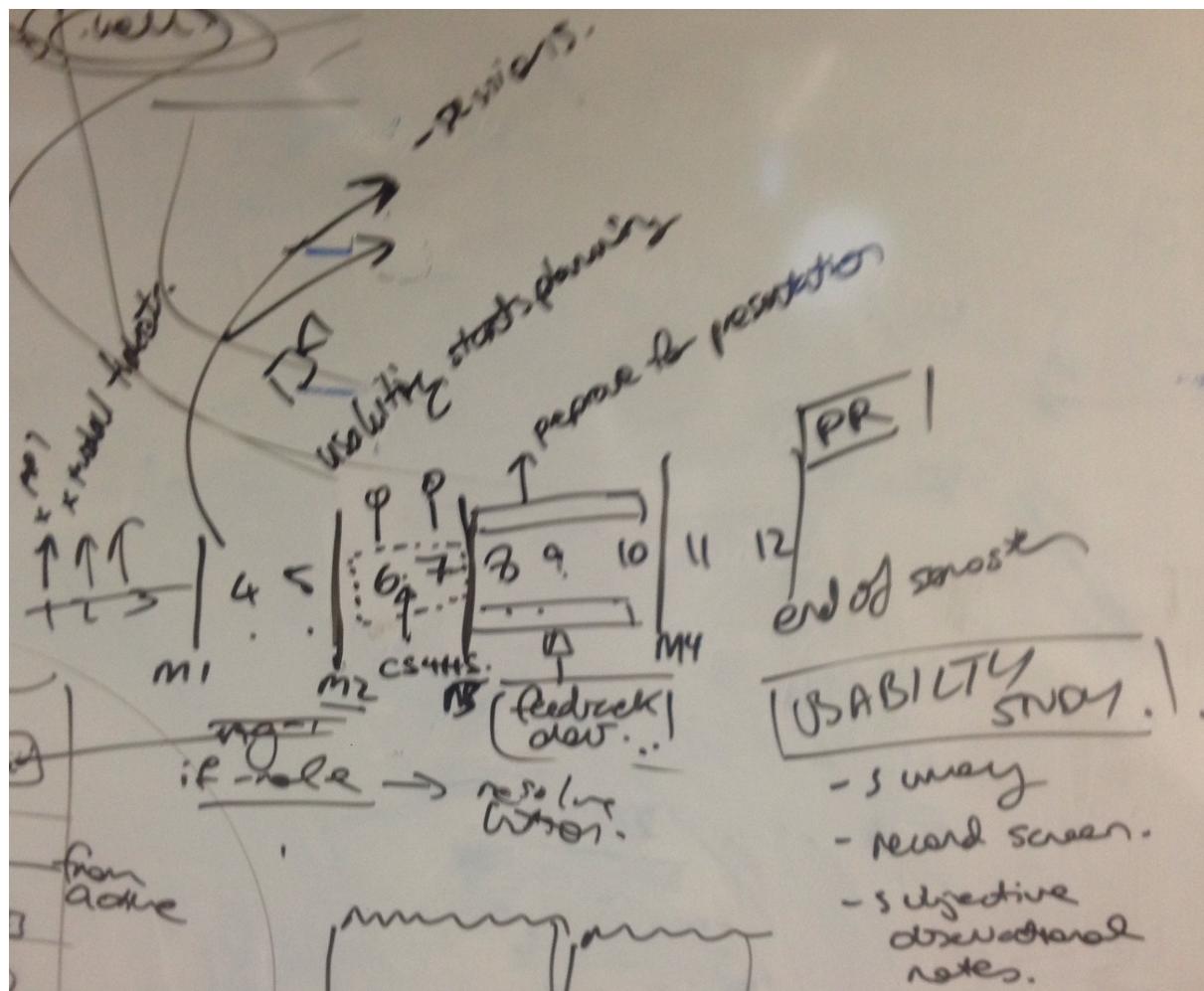


Figure 4: Sketch of timeline on whiteboard

4.4.1 Follow Up Actions

1. **Alex - Next week** - Make a legible timeline out of whiteboard and print out for next week
 2. **Alex - Next week** - Make barebones foundation for Lachlan and Reuben to work with and improve
 3. **Jake, Alex - 2 weeks** - Work on wireframe of dashboard UI
-

5 Meeting 30 Aug 12:00pm

Location: ATC Project Rooms

5.1 Present

- Jake Renzella
- Lachlan West
- Reuben Wilson

5.2 Agenda Items

1. Review Sprint Plan
2. Review Changes to Codebase

5.3 Discussion, Decisions and Agreements

5.3.1 RE: 1. Review Sprint Plan

- Team reviewed the sprint plan
- Agreed that dates were good and achievable

5.3.2 RE: 2. Review Changes to Codebase

- Codebase changes around API and Web
- Testing is 100% done on API with all endpoints - code is ready to be locked
- Front end is set up with all requests being made to the API

- Only thing outstanding is the dashboard stats chart
 - e.g. how many people are in help desk. How many tickets etc.
 - Alex still do to this along with additional changes to stats

5.3.3 Additional Items

- Progress review is due next week
- Film stock footage of us coding next week

5.4 Agenda Items For Next Meeting

1. Handover wireframe to Reuben and Lachlan
2. Progress on UI changes

5.4.1 Follow Up Actions

1. **Reuben and Lachlan - ASAP** - Take what Alex has done and create proper looking Ticket creation. CSS and logic such as ticket expanding
 2. **Jake - Next week** - Do wireframe, collaborating with Reuben & Lachlan
 3. **All - Next week** - Peer reviews
-

6 Meeting 6 Sept 12:00pm

Location: ATC Project Rooms

6.1 Present

- Jake Renzella
- Lachlan West
- Alex Cummaudo

6.2 Agenda Items

1. Progress on dashboard framework
2. Usability evaluation test planning

6.3 Discussion, Decisions and Agreements

6.3.1 RE: 1. Progress on dashboard framework

- Alex showed progress made on dashboard
- Agreed that the layout could be rearranged
- Discussed what options were available for the dashboard
 - Placing of stats at top
 - Placing of queue at bottom
 - Graph and tutors can be re-arranged

6.3.2 RE: 2. Usability evaluation test planning

- Usability preparation needs to happen soon
- Plan on executing evaluations Week 7 (after break)
- What is the purpose of the document?
 - Usability testing on the helpdesk dashboard
 - Benchmark of current system
 - Run survey to get information on what they have now
 - Show them a prototype
 - Run that survey again based on their experience of that prototype
- What kinds of questions do we ask
 - Background:
 - * Do you work at the helpdesk or study there?
 - * How many hours?
 - * Age range
 - * If you were at the back of the helpdesk, can you read writing on projector?
 - * Different font types and sizes
 - * List the three top things you would expect to see on a dashboard at the Helpdesk
 - * If staff:
 - On average how many hours do you work in one shift
 - How long have you been working (semesters)

- What subject(s) do you teach?
 - How many students to concurrently support without being overburdened?
 - How long do you think is acceptable for students to wait?
 - List 3 bits of information would be most helpful for you before you see a student?
- * If student:
- What is the max time you would be happy waiting for?
 - How long on average do you have to wait now?
 - If you could see how busy the helpdesk is before going there, would it affect the likelihood of going there.
 - Before going to the helpdesk what information would be helpful before going
- After showing them the prototype:
- * List three things you liked
 - * List three things you didn't like (what would you change)
 - * Would you go to the helpdesk more if you had this
 - * Could you read all information? If not what couldn't you read?
 - * Would you prefer to sign in to access this information?
 - * What more information would you want to see?
 - * Would you like to be notified when a ticket is resolved by an: audio cue, visual cue, both, none
 - * Any additional comments

6.3.3 Additional Items

- Peer reviews are due this week!
- Try and get work logs up to date for this week

6.4 Agenda Items For Next Meeting

1. Review evaluation details made

6.4.1 Follow Up Actions

1. Alex - **ASAP** - Improve dashboard to match Jake's wireframes
 2. Jake and Lachlan - **Next week** - Work on usability docs
 3. All - **Next week** - Film some presentation video footage
-

7 Meeting 20 Sept 10:15am

Location: Graham's Office

7.1 Present

- Jake Renzella
- Alex Cummaudo
- Graham Farrell

7.2 Agenda Items

1. Progress update

7.3 Discussion, Decisions and Agreements

7.3.1 RE: 1. Progress update

- Graham is happy with progress so far, has been discussing with Andrew
- Suggests to focus priority on:
 1. Finishing Usability Evaluation
 2. Presentation video for Week 11
 3. Bundling together work into portfolio for Week 12
- Code lock as soon as possible

7.3.2 Additional Items

None

7.4 Agenda Items For Next Meeting

1. Progress update

7.4.1 Follow Up Actions

1. **Alex and Jake - Next 2 weeks** - Modify last semester's presentation to include latest work
 2. **Lachlan - Next week** - Finalise changes to web codebase
 3. **All - ASAP** - Film testimonials of system in use for presentation
-

8 Meeting 4 October 11:15am

Location: ATC621

8.1 Present

- Jake Renzella
- Alex Cummaudo
- Reuben Wilson
- Lachlan West

8.2 Agenda Items

1. Planning for video
2. Compiling Portfolio
3. Finish usability document
4. Who did what document

8.3 Discussion, Decisions and Agreements

8.4 RE: 1. Planning For Video

- Filming to be done Tuesday 6th October at helpdesk 1:30pm
- Film four sections:

- What is the helpdesk? (30-40s)
 - Reuse stock footage from last semester
 - Tim's testimonial of what is wrong
 - Fallback's were Sublime Text windows on projector: What was wrong with that?
 - * 1. It was hard to assign tutors to each person
 - * 2. It was impossible to get statistics
 - * 3. Text based means non-dynamic and barely interactive
- Analysis of problem
 - Helpdesk Ticketing System
 - Integrate this into a common system that already exists and is used: DF
 - That way people don't have to relearn another system from scratch
 - Additional changes to DF
 - Make it possible to attain statistics
 - Wireframes:
 - All accessible from the Helpdesk Menu in DF
 - Handle both mobile-web app and dashboard web app
 - Show wireframes on whiteboard
- Prototype
 - Student
 - Run through create a ticket scenario
 - Dashboard
 - Ticket displays on the dashboard
 - * show ticket being created
 - Graph changes and updates dynamically based on
 - * show numbers updating
 - * show graph changing
 - Staff using system
 - * staff clicking onto ticket on their own machine
 - * show someone pretending to help the student

- * show staff resolving the ticket
 - * show similar situation on mobile
- Staff clock on/off
- Usability
 - Ran through a usability study on the system with students
 - Show cliff photos using HD
 - People doing evaluation
 - Discuss what people noted: quotes on screen
- Testimonials
 - Footage of Charlotte/Adrian talking about the HD
- Future
 - Talking about discussion on improving it in the future
 - It's open source so help us out!

8.5 RE: 2. Compiling portfolio

- Code lockdown as of now

8.6 RE: 3. Finish usability document

- Reuben, Jake and Lachlan are working on this document

8.7 RE: 4. Who did What document

- All must fill out each of their respective sections ASAP

8.7.1 Additional Items

None

8.8 Agenda Items For Next Meeting

1. Progress update

8.8.1 Follow Up Actions

1. **Alex - ASAP** - Try and get a tripod
2. **Jake - ASAP** - Upload videos to cloud
3. **Reuben, Lachlan, Jake - ASAP** - Finish Usability Report
4. **Alex - Week 12** - Finish compiling portfolio
5. **Alex - ASAP** - Download code changes as of today

Image Attributions

‘Scrum Development Team’ designed by Roy Verhaag

‘Storage’ designed by Dirk-Pieter van Walsum

‘Technical Requirements’ designed by Alvarobueno

‘Quality Assurance’ designed by Lorenzo Stella

‘Agile’ designed by Nimal Raj

‘Concept Prototype’ designed by Yu Luck

‘LOG File’ designed by Viktor Vorobyev

‘Meeting’ designed by Marie Van den Broeck

‘Plan’ designed by Michael Finney

‘Profile User Flow’ designed by Sharon Showalter

From The Noun Project collection

<http://www.nounproject.com>