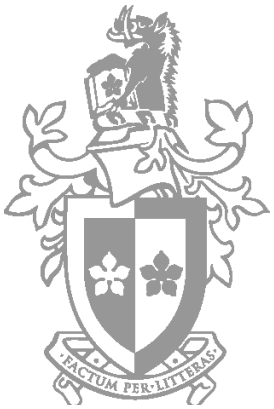


Doubtfire Helpdesk Ticketing System

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 LaTeX using Pandoc. Should there be formatting issues, please refer to the original content on the GitHub wiki instead.

Declaration

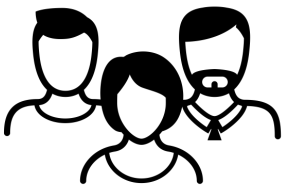
We declare that this portfolio is a group assessment and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for us by another person.

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

Contents

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

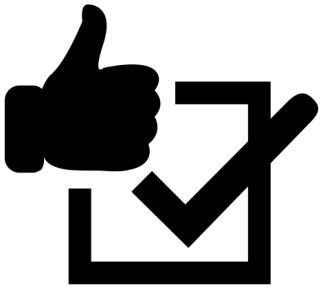
Student Number: 2423480
Email: 2423480@student.swin.edu.au
GitHub: lachlanwest
Roles:

- 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

- [x] Cover Sheet
- [x] Checklist
- [ ] Research Report[1]
- [x] Updated[2] Project Plan
- [x] Updated[3] SDLC Plan
- [x] Updated[4] Requirements Specification Document
- [x] UI Prototypes
- [x] Usability Testing Report
- [x] Technical Manual
- [x] User Manual
- [ ] Table of Contents of Lab Manual[5]
- [x] Source Code Directory Dump
- [x] API + Web Source Code Changes
- [ ] MoU[6]
- [x] CD - Source Code
- [x] 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)
   - API Changes (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

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 documents[1] 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.

[1] See `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.

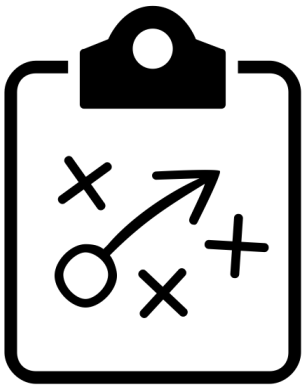
Such a grade would be awarded to a group member who has:

- **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

1.1 Purpose of this document

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

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

1.2 Background

1.2.1 Swinburne University Programming Helpdesk

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

1.2.2 Doubtfire Learning Management System

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

1.3 Key Project Personnel

1.3.1 Client

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

1.3.2 Stakeholders

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

1.3.3 Project Manager

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

1.3.4 Project Members

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

2 Terms of Reference

2.1 Goals

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

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

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

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

2.2 Objectives

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

2.3 Scope

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

2.4 Critical Success Factors

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

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

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

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

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

2.5 Acceptance Criteria

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

2.5.1 Convenors

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

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

2.5.2 Students

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

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

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

2.5.3 Tutor

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

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

3 Establishment

3.1 Process, Procedures and Standards

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

3.2 Project Environment

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

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

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

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

Using the system requires:

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

3.3 Project Team Skills Requirements

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

3.3.1 API Server

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

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

3.3.2 Web Interface

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

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

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

3.3.3 Additional knowledge

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

4 Activities, Deliverables and Capital Resources

4.1 Deliverables

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

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

4.2 Activities and Tasks

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

4.2.1 Research

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

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

4.2.2 Development

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

4.2.3 Testing

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

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

4.2.4 Documentation Authoring

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

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

4.2.5 Submission for Critical Analysis

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

4.3 Resources

4.3.1 Organisation Structure

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

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

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

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

4.3.2 Development Equipment

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

5 Risk Analysis

5.1 Risk 1

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

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

5.2 Risk 2

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

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

5.3 Risk 3

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

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

5.4 Risk 4

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

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

5.5 Risk 5

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

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

5.6 Risk 6

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

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

5.7 Risk 7

Possible Issue: Document availability issues
Chance of Occurrence: Low
Severity of Detriment to Group Ambitions: Severe
Preventative Methods:

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

5.8 Risk 8

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

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

5.9 Risk 9

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

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

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

6 Schedule

6.1 Overview and Timeline

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

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.

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

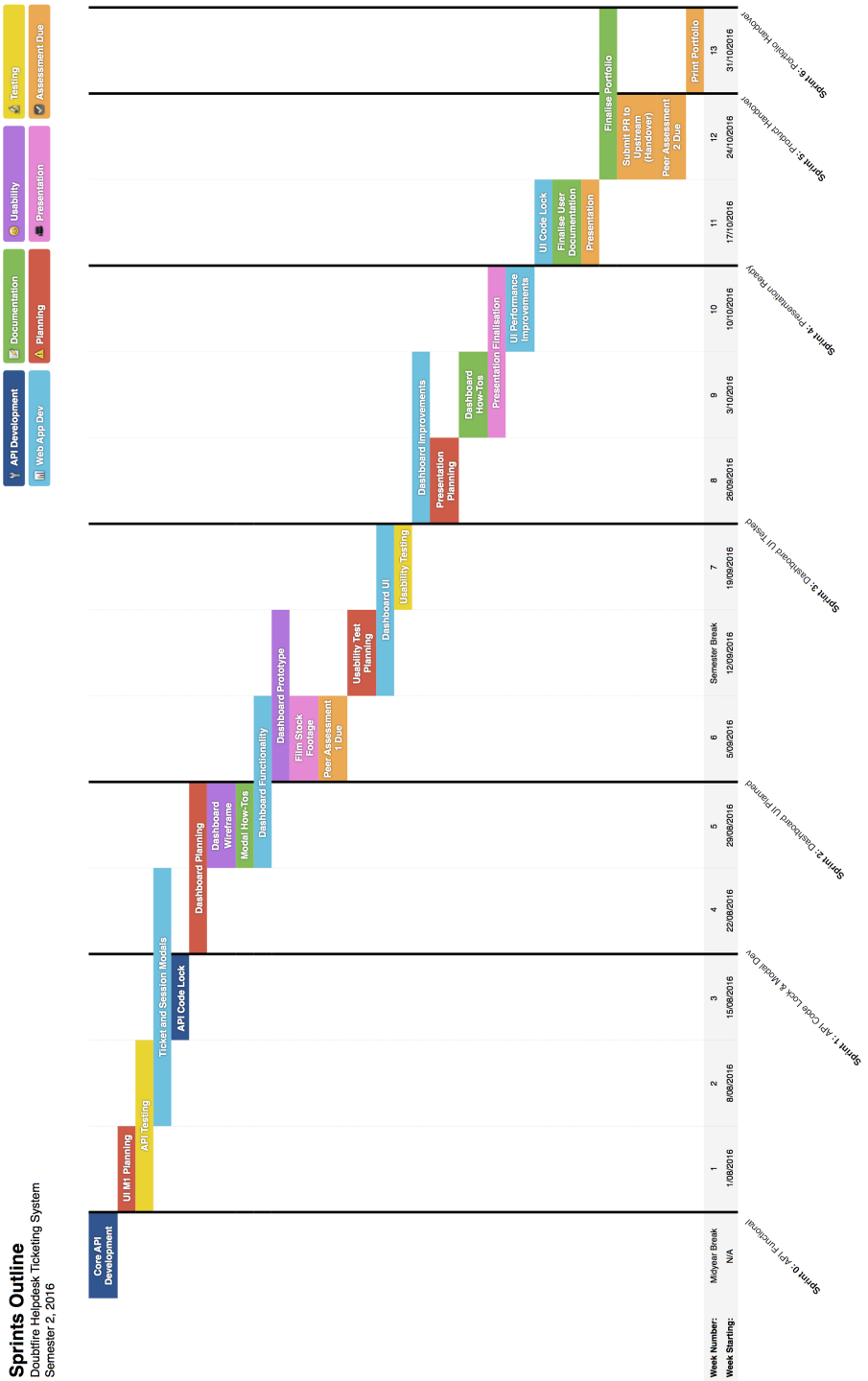


Figure 1: Sprint Plan for Semester 2

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.

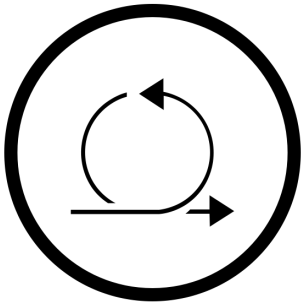
Estimates for the project have been calculated as thus:

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

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 meeting[1] 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.

[1] See `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.

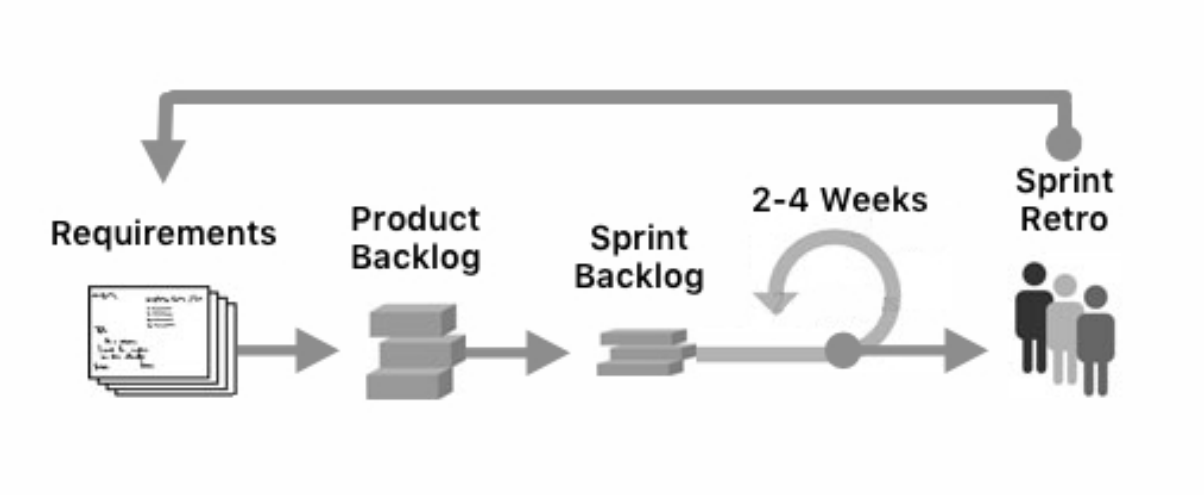


Figure 1: High level overview of the workflow

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 Tools[3] 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 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

[2] See `https://en.wikipedia.org/wiki/Continuous_integration`

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

[4] See `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:

[5]See `https://trello.com/b/8a1k0Wud/helpdesk-ticketing-system`

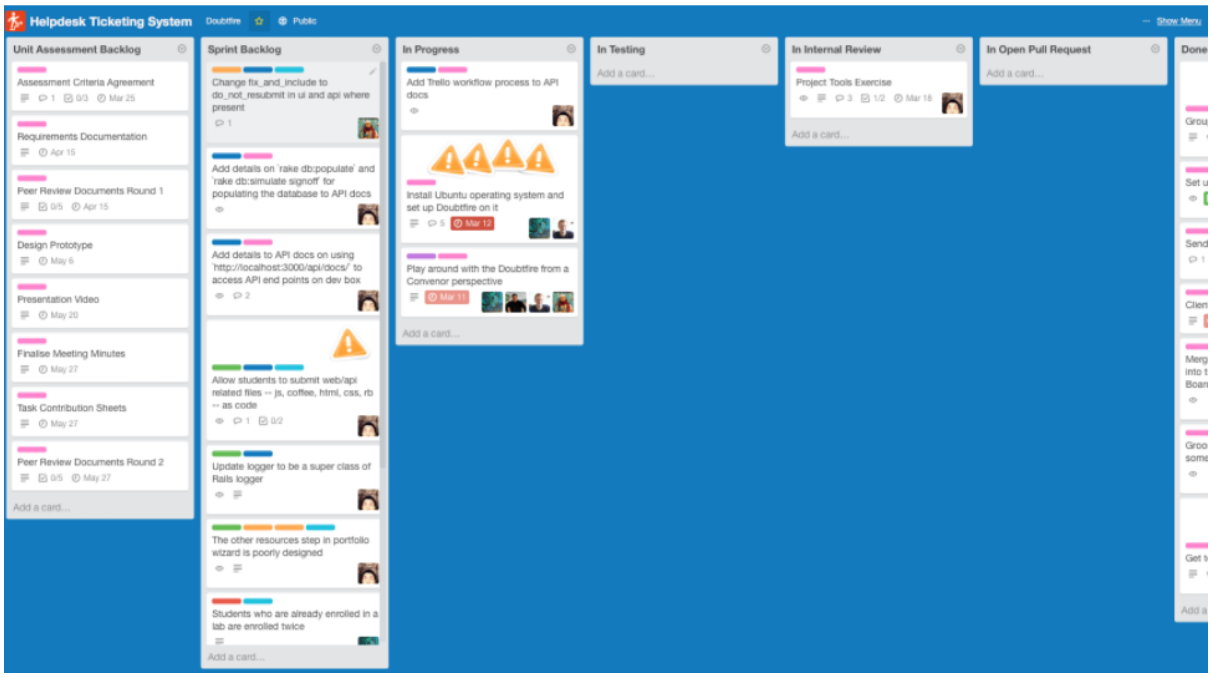


Figure 3: The Helpdesk Ticketing System Board

1. **Unit Assessment Backlog** - The backlog of tasks that are required by the final year project unit deliverables[6]
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**.

[6]See `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 Workflow[7], 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 Board[8] 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 Board[9]'s Sprint Backlog column. To do this, they can click on a card and them 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:

[7]See `https://github.com/doubtfire-lms/doubtfire-api/blob/develop/CONTRIBUTING.md`
[8]See `https://trello.com/b/0uh6AZdu/doubtfire-backlog`
[9]See `https://trello.com/b/8a1k0Wud/helpdesk-ticketing-system`

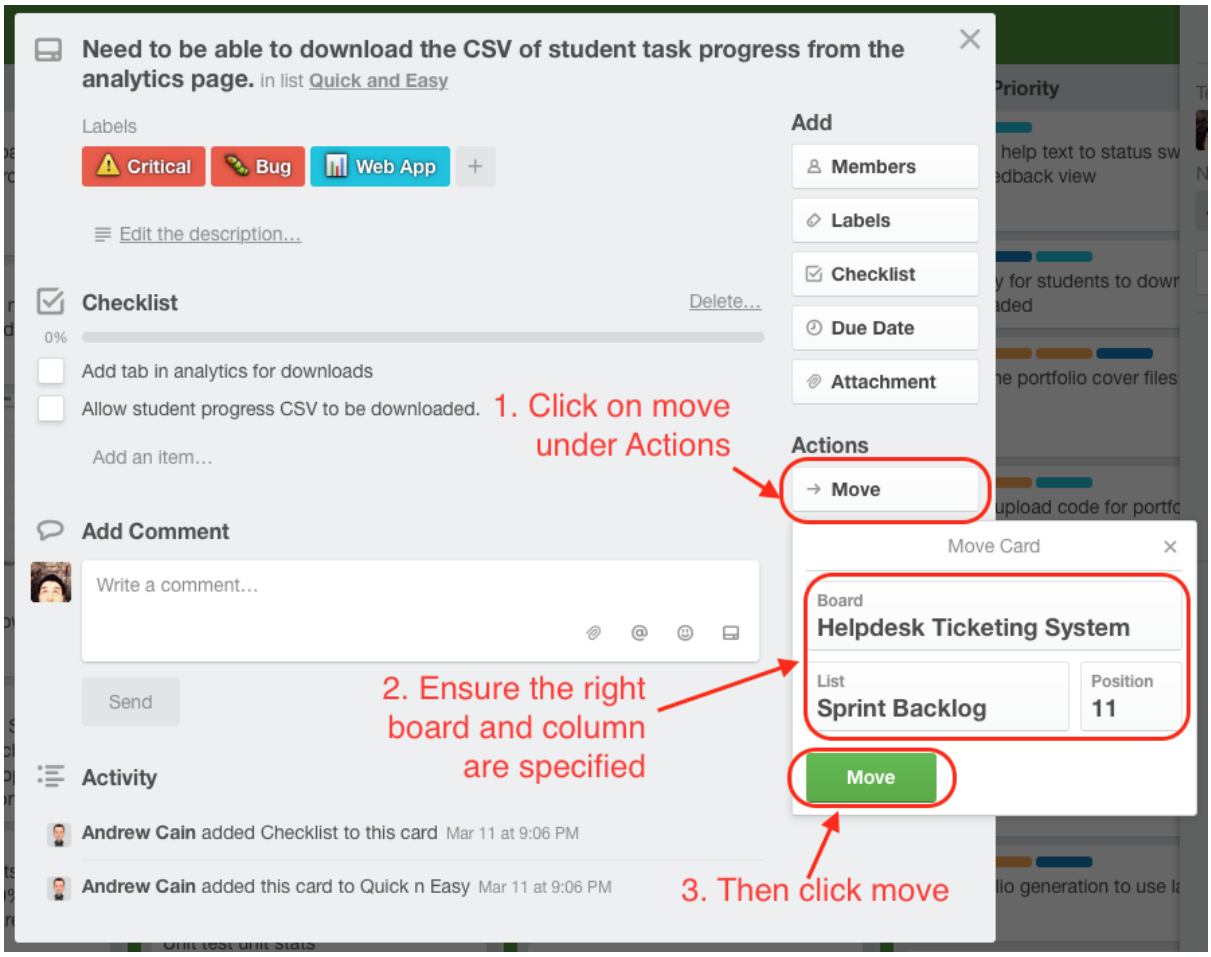


Figure 4: Moving a card to the right board

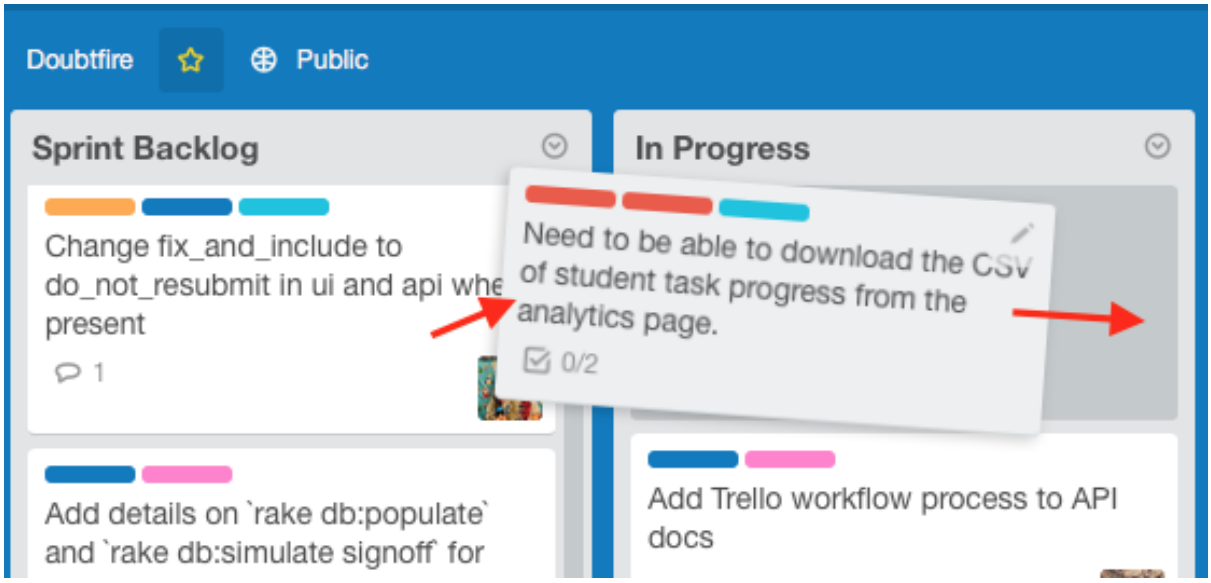


Figure 5: Dragging and dropping the card

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 Workflow[10] a Pull Request should be submitted to the

[10]See `https://github.com/doubtfire-lms/doubtfire-api/blob/develop/CONTRIBUTING.md#`

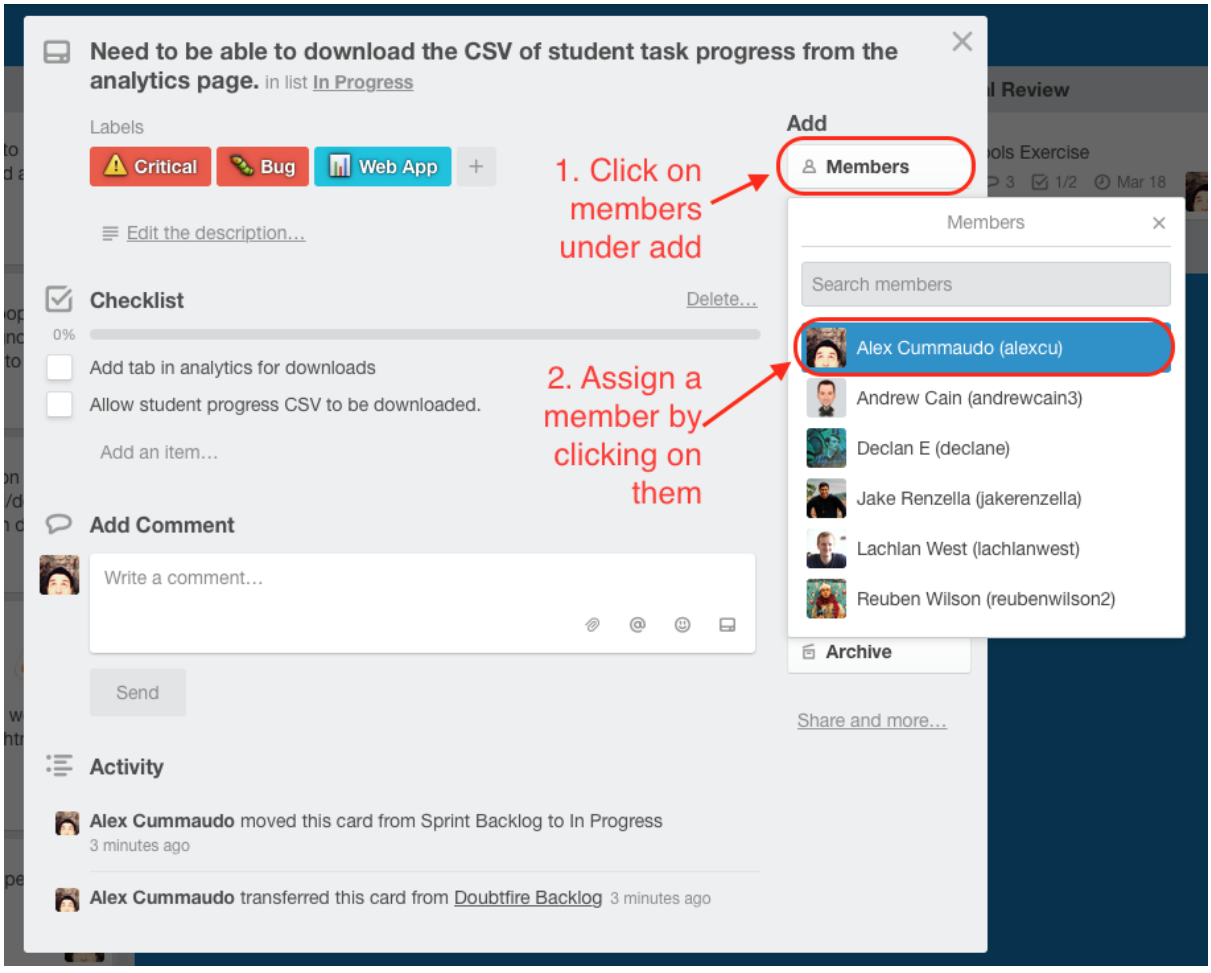


Figure 6: Assign a member

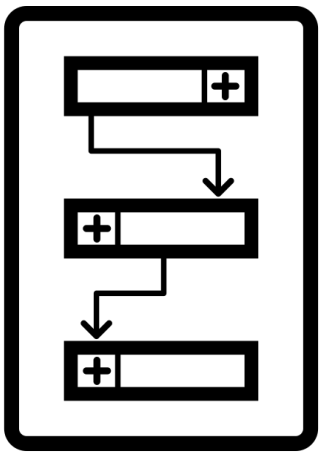
Product Owners (i.e., the `doubtfire-lms` repository) when ready for external review.

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

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 to see how their tutors are clocking on at the helpdesk

Refer to the requirements documentation for more on this.

2 Entities

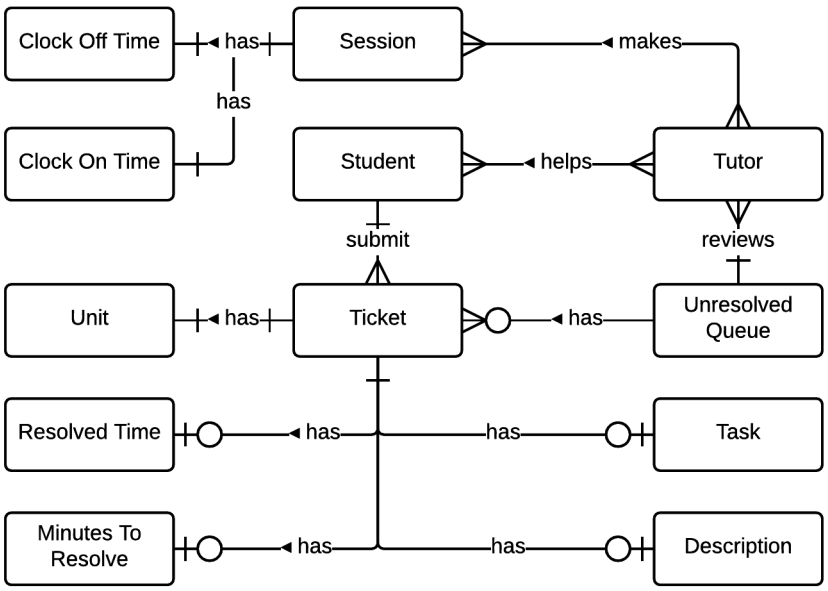


Figure 1: ERD

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

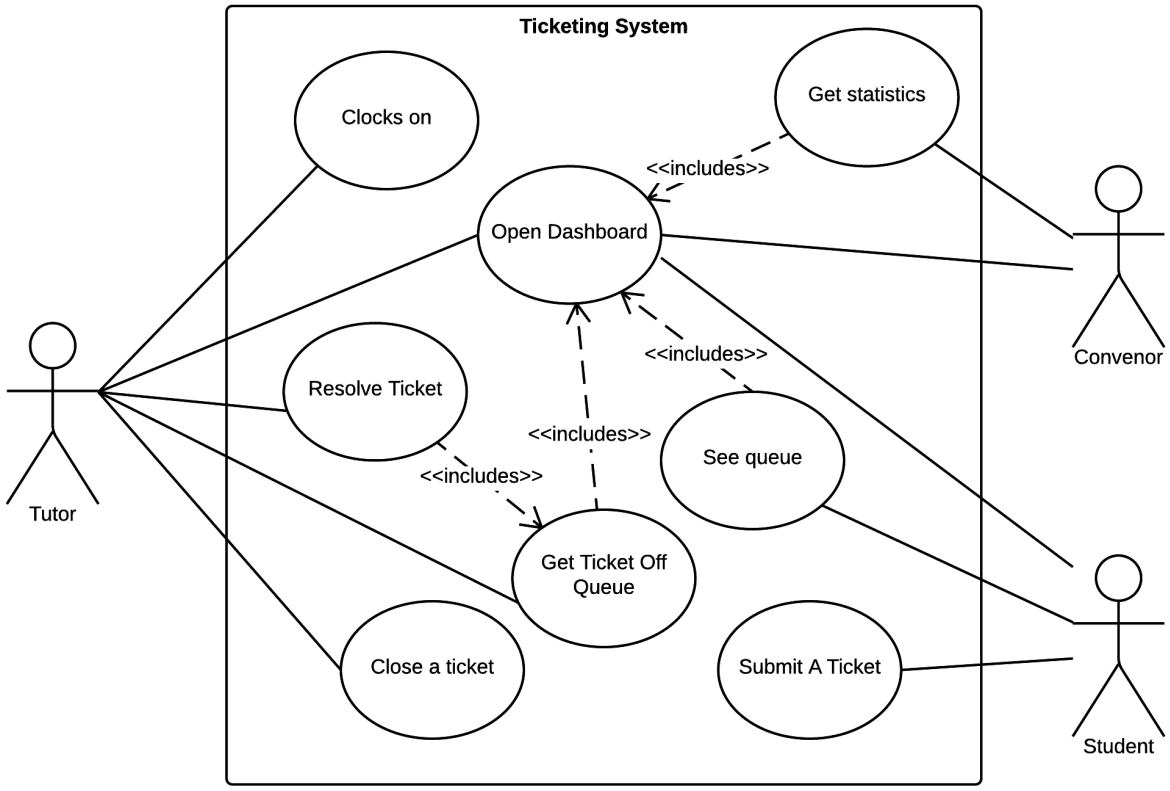


Figure 2: Use Case Diagram

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

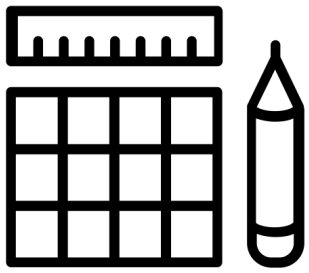
3.2.2.1 Primary Use Case

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

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

1.1 Iteration One

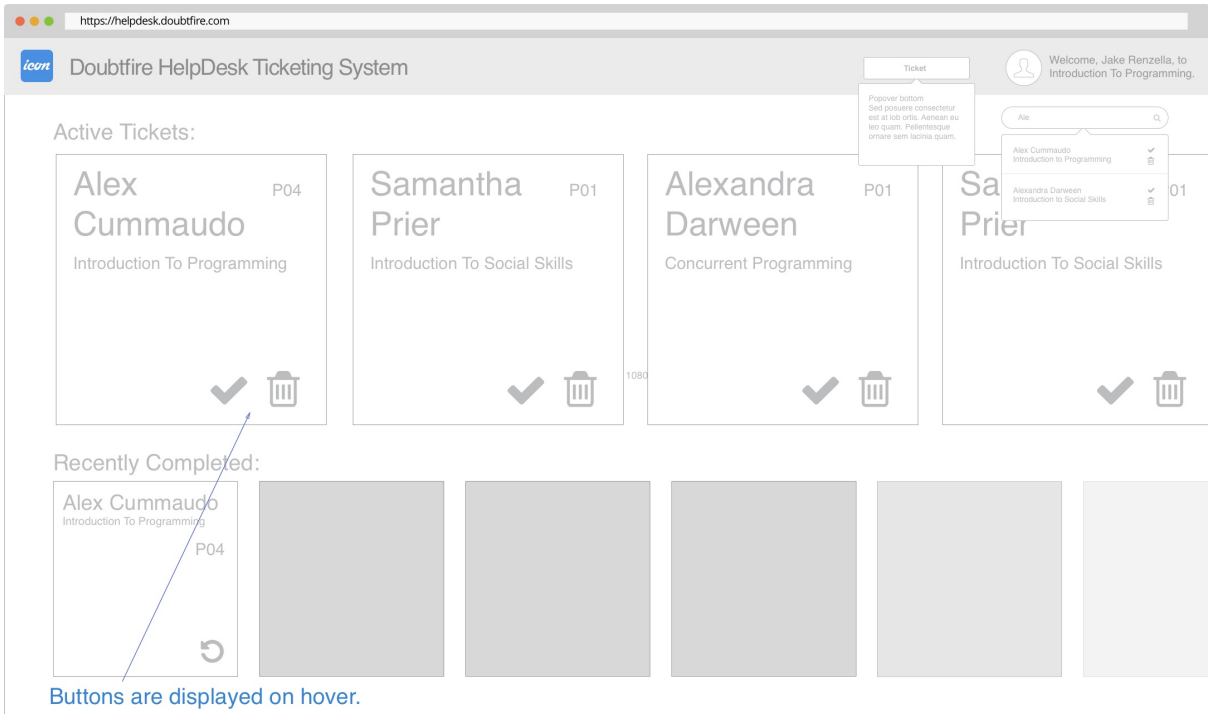
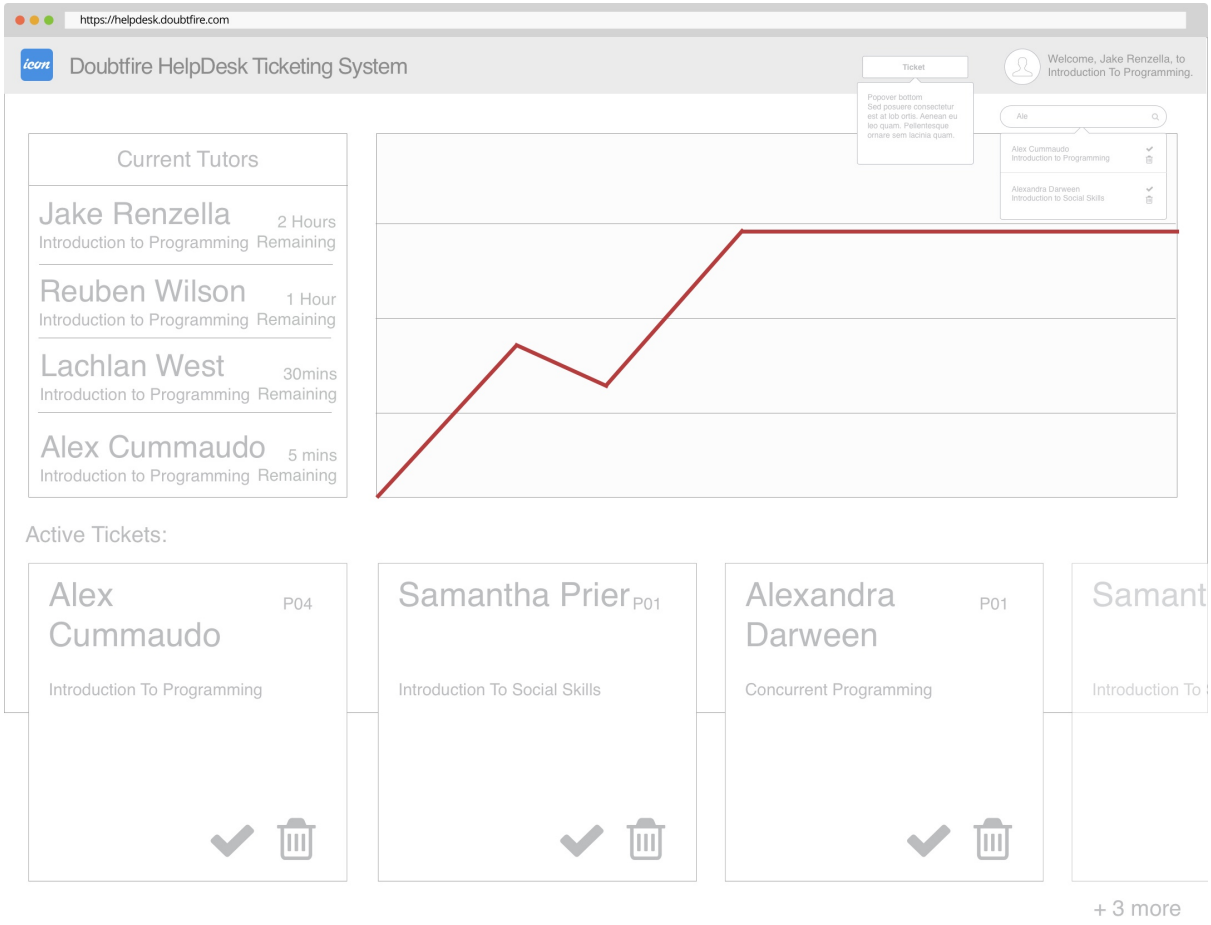


Figure 1: Iteration One of the Dashboard

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).

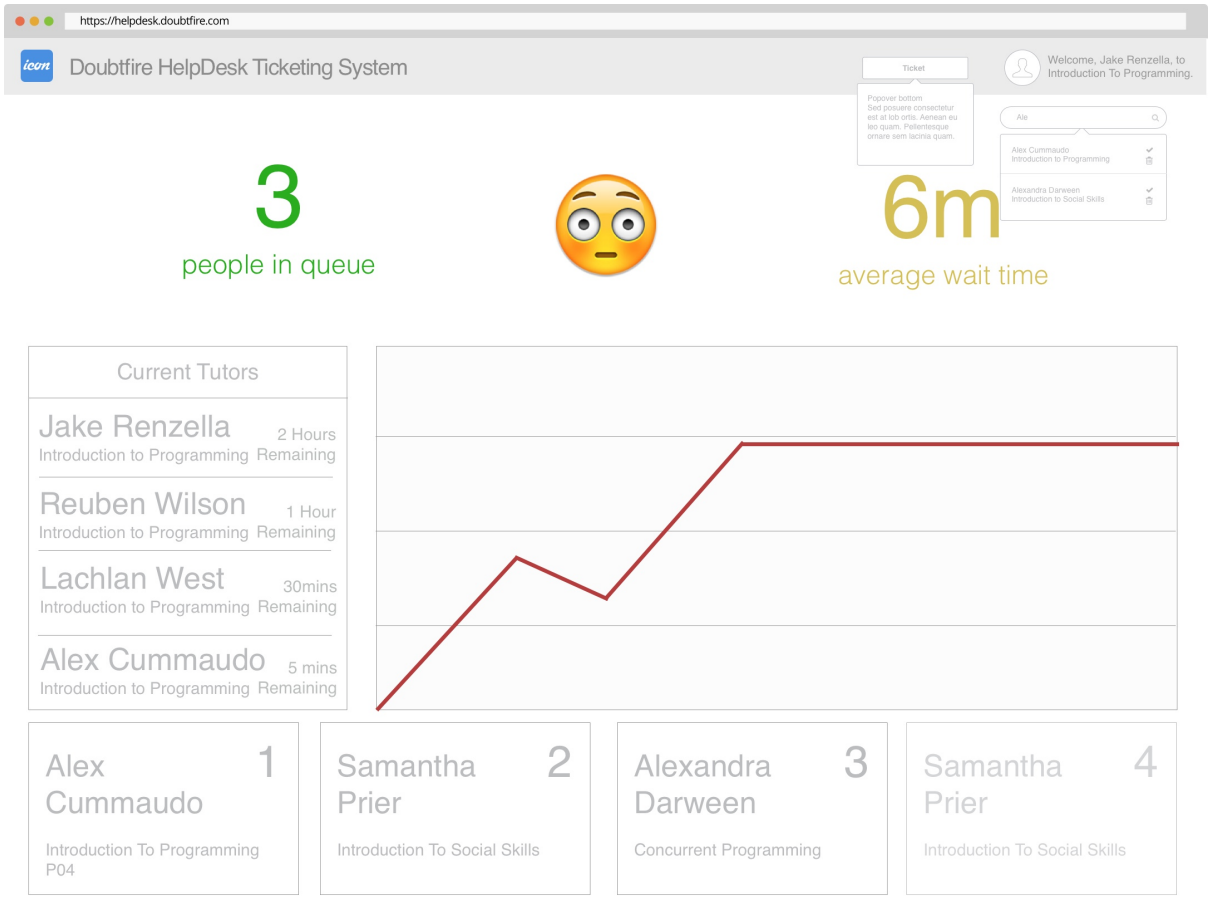
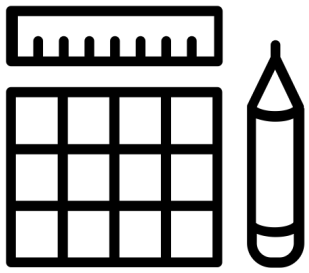


Figure 2: Iteration Three of the Dashboard



Chapter 8

Usability Testing Report

Contents

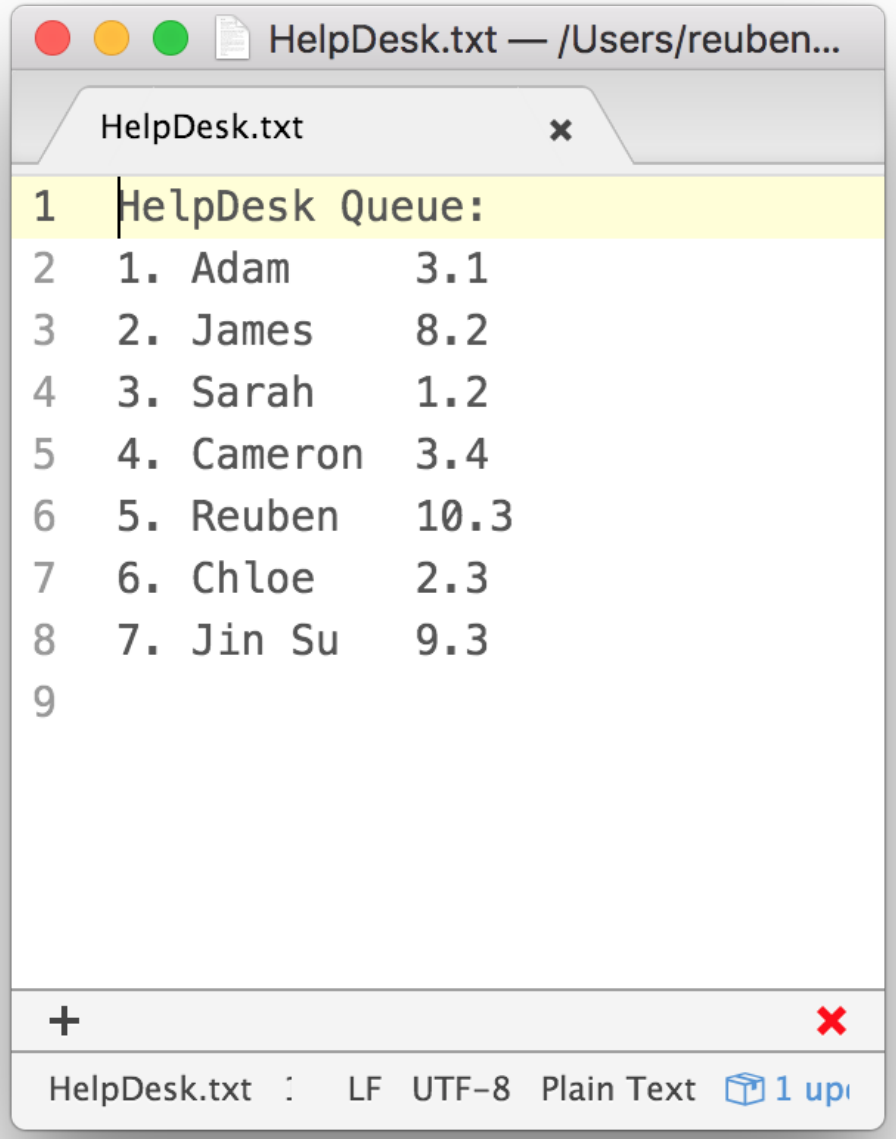


Figure 1: Previous Helpdesk System

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.

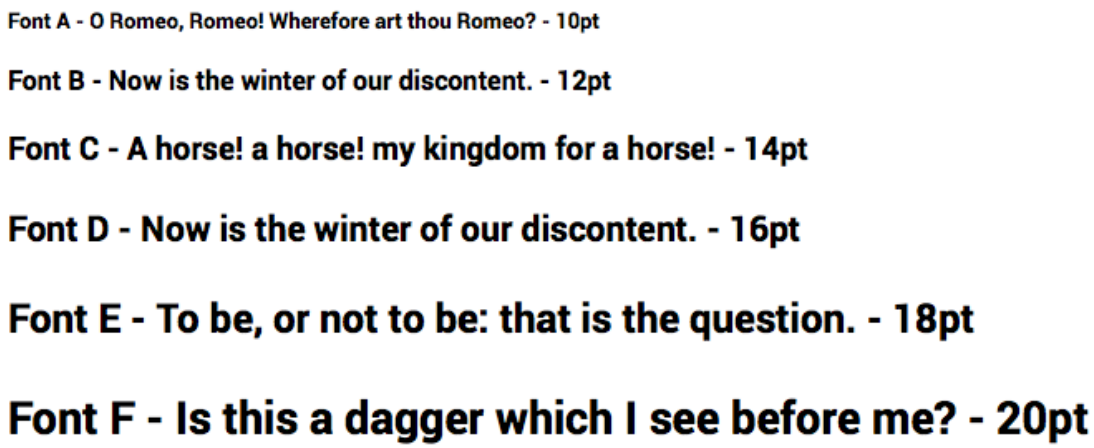


Figure 2: Font size test

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 if 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				
Font B - 12pt				
Font C - 14pt				
Font D - 16pt				
Font E - 18pt				
Font F - 20pt				

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 familar	◯	◯	◯	◯	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.	◯	◯	◯	◯	◯
I found the system unnecessarily complex.	◯	◯	◯	◯	◯
I thought the system was easy to use.	◯	◯	◯	◯	◯
I think that I would need the support of a technical person to be able to use this system.	◯	◯	◯	◯	◯
I found the various functions in this system were well integrated.	◯	◯	◯	◯	◯
I thought there was too much inconsistency in this system.	◯	◯	◯	◯	◯
I would imagine that most people would learn to use this system very quickly.	◯	◯	◯	◯	◯
I found the system very cumbersome to use.	◯	◯	◯	◯	◯
I felt very confident using the system.	◯	◯	◯	◯	◯
I needed to learn a lot of things before I could get going with this system.	◯	◯	◯	◯	◯

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	◯	◯	◯	◯
Task 2	◯	◯	◯	◯
Task 3	◯	◯	◯	◯
Task 4	◯	◯	◯	◯
Task 5	◯	◯	◯	◯

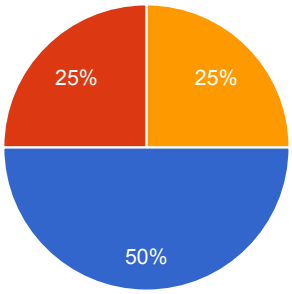
C Raw Results

See attached document

Results

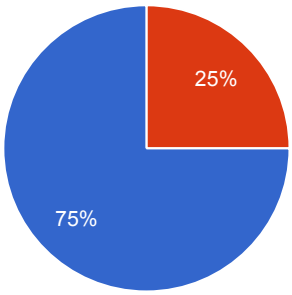
Demographics

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%

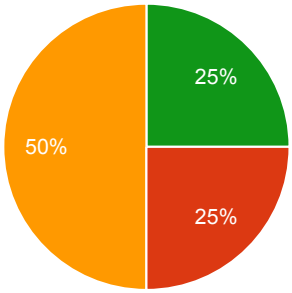
Select your gender.



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

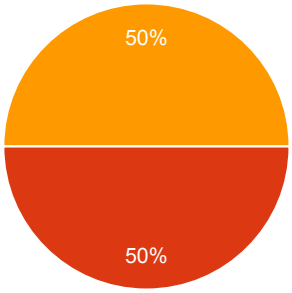
Helpdesk Attendance

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



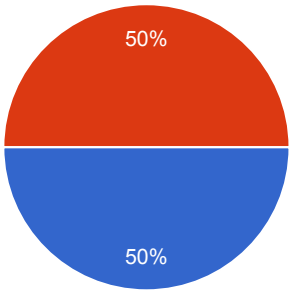
More than four hours 1 25%

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%

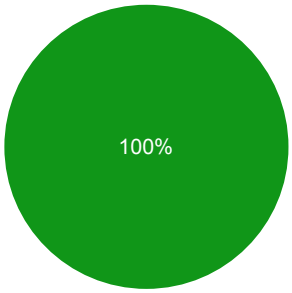
Do you attend the helpdesk as a staff member or as a student?



A staff member	2	50%
A student	2	50%

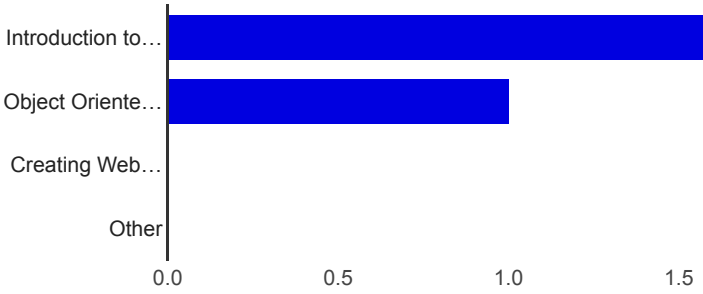
Staff Questions

How many semesters have you taught at the helpdesk?



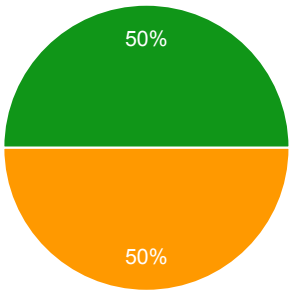
One semester	0	0%
Two to three semesters	0	0%
Three to four semesters	0	0%
More than four semesters	2	100%

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%

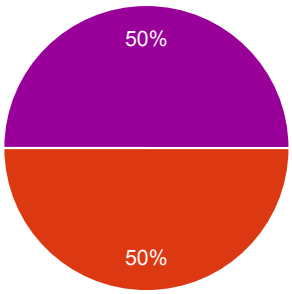
How many students to concurrently support without being overburdened?



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

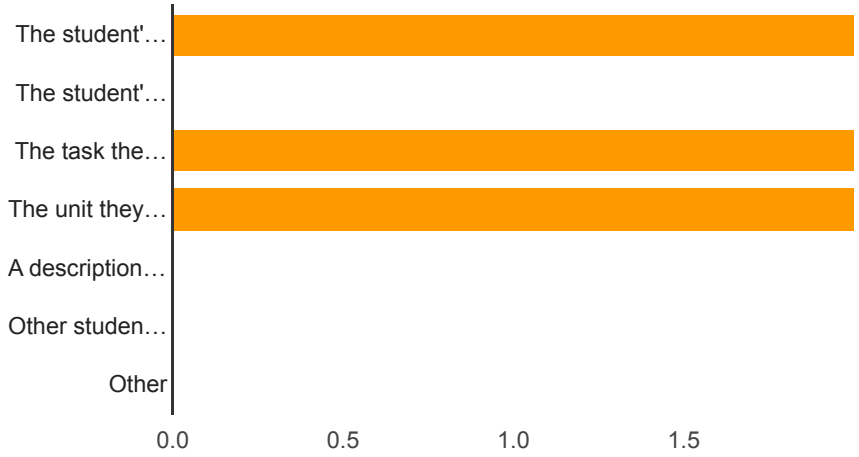
More than five students	1	50%
More than six students	0	0%

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%

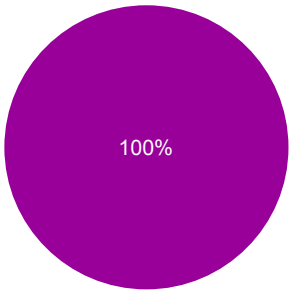
What three pieces of information would be most helpful for you before you see a student for help?



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%

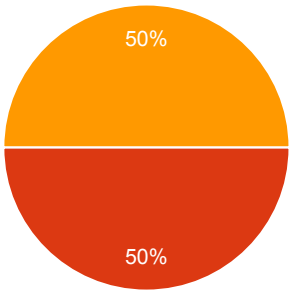
Student Questions

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



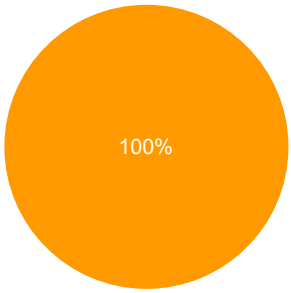
More than two minutes	0	0%
More than three minutes	0	0%
More than four minutes	0	0%
More than five minutes	0	0%
More than six minutes	2	100%

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



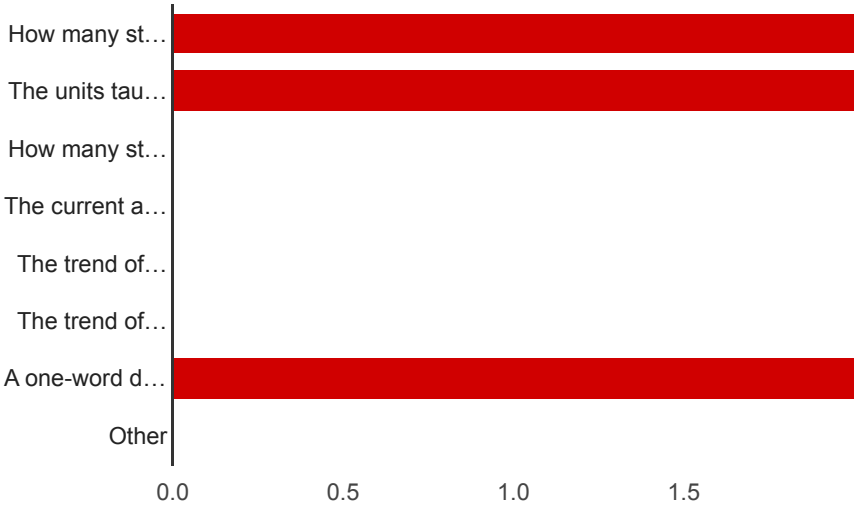
More than two minutes	0	0%
More than three minutes	1	50%
More than four minutes	1	50%
More than five minutes	0	0%
More than six minutes	0	0%

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



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

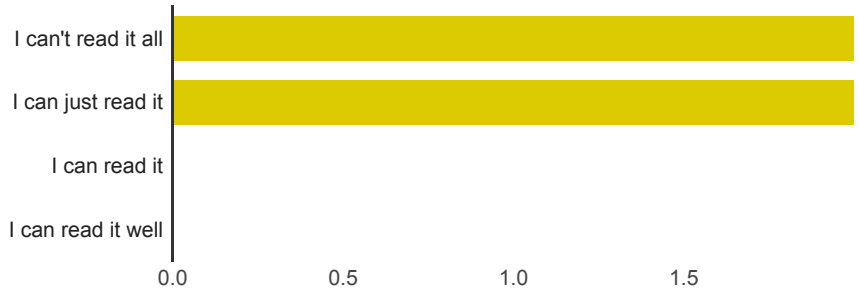
Before going to the helpdesk, what three pieces of information would be helpful before going?



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%

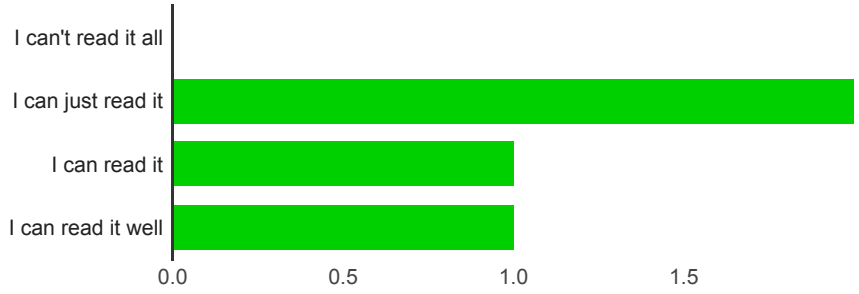
Font Testing

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



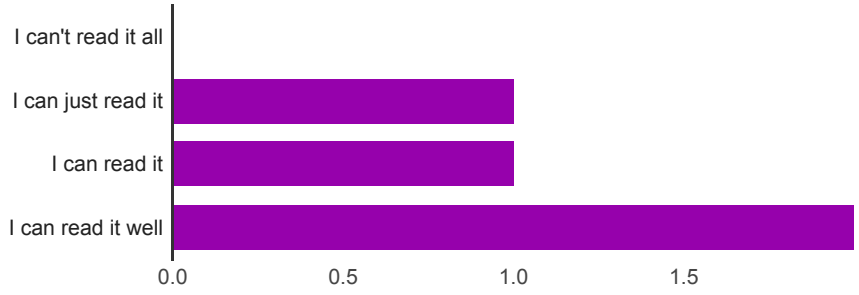
I can't read it all	2	50%
I can just read it	2	50%
I can read it	0	0%
I can read it well	0	0%

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%

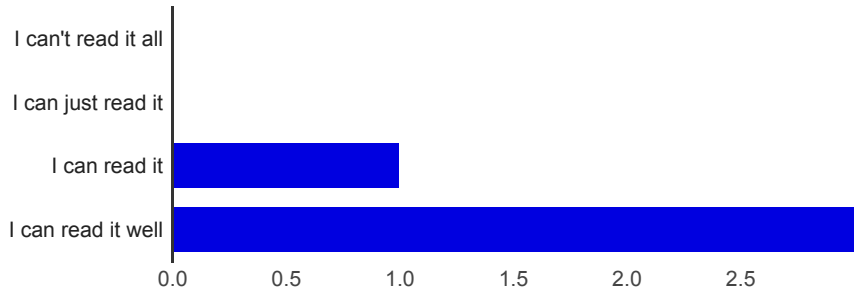
Font C - 14pt [Rate each font on the screen based on how well you could read the sentence.]



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%

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

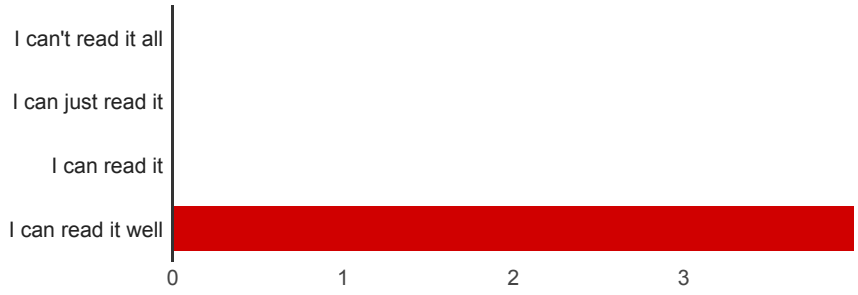


I can't read it all	0	0%
I can just read it	0	0%
I can read it	1	25%
I can read it well	3	75%

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

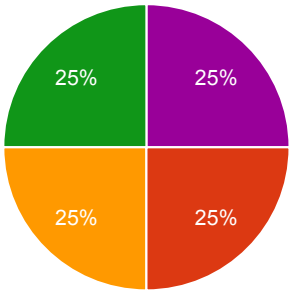
I can't read it all	0	0%
I can just read it	0	0%
I can read it	0	0%
I can read it well	4	100%

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



I can just read it	0	0%
I can read it	0	0%
I can read it well	4	100%

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 whats working

Graph, tutors are on the lefthand side

List two things you would change about the dashboard.

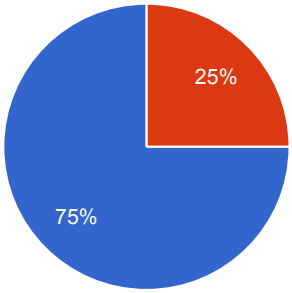
areas could be more well defined, average wait time graph might be unnecessary

dont 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?

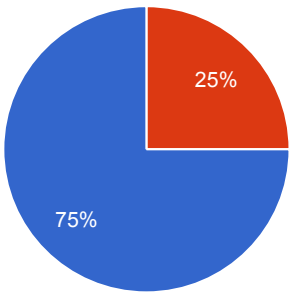


Yes	3	75%
No	1	25%

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

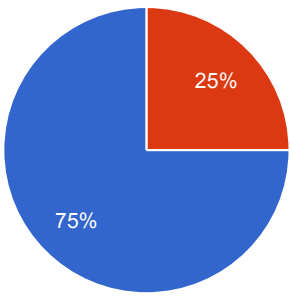
Axis on graph

Could you interpret what the graph was displaying?



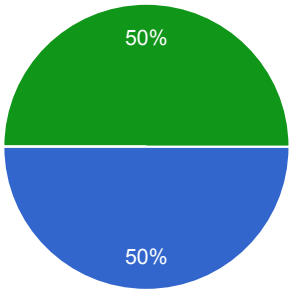
Yes	3	75%
No	1	25%

Would you prefer to sign in to access this information?



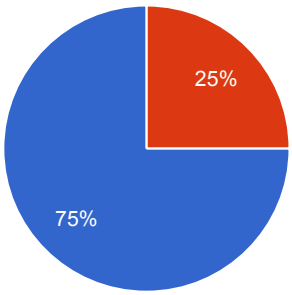
Yes	3	75%
No	1	25%

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



None 2 50%

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%

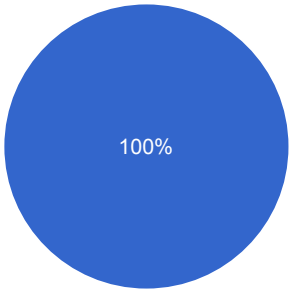
Please note any additional comments you would like to make about the dashboard you have been shown.

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%

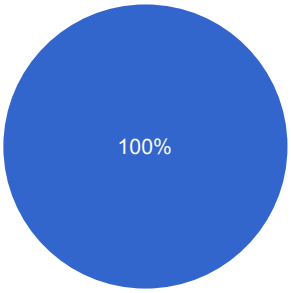
User Task Descriptions

Task 1

Task Description

Information Needed

Were you able to successfully create your ticket?



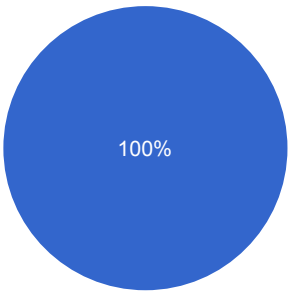
Yes	4	100%
No	0	0%
I don't know	0	0%

After finishing this task...

What is your ticket number in the queue?

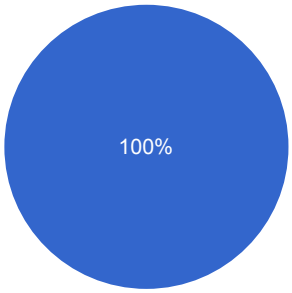
4

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



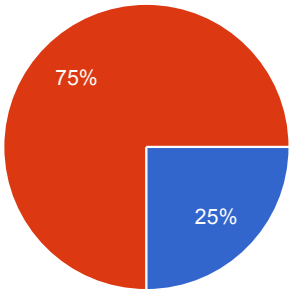
Yes	4	100%
No	0	0%
I don't know	0	0%

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?

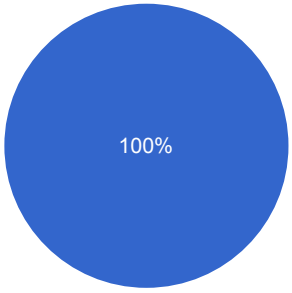


Zero to one hours ago	1	25%
One to two hours ago	3	75%
Two to three hours ago	0	0%

Task 2

Task Description

Were you able to close your ticket successfully?



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

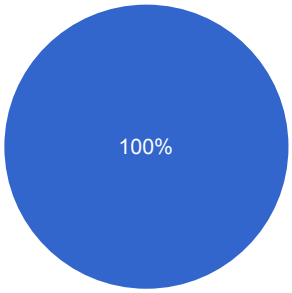
After finishing this task...

Task 3

Task Description

Information Needed

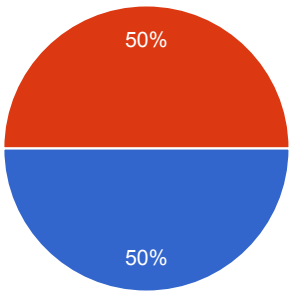
Were you able to clock on at the Helpdesk for the scheduled 2 hours and 45 minutes?



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?



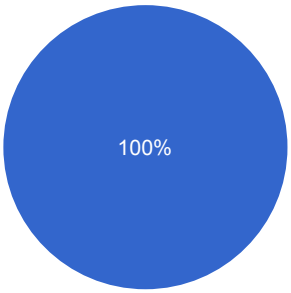
Yes	2	50%
No	2	50%
I don't know	0	0%

Task 4

Before you begin...

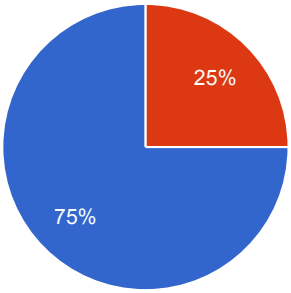
Task Description

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?

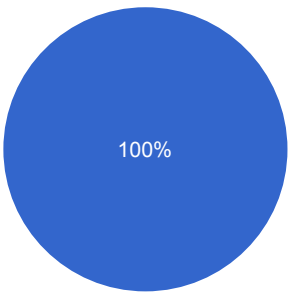


Yes	3	75%
No	1	25%
I don't know	0	0%

Task 5

Task Description

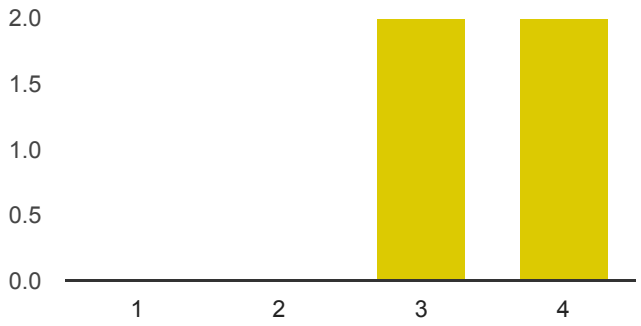
Were you able to clock off successfully?



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

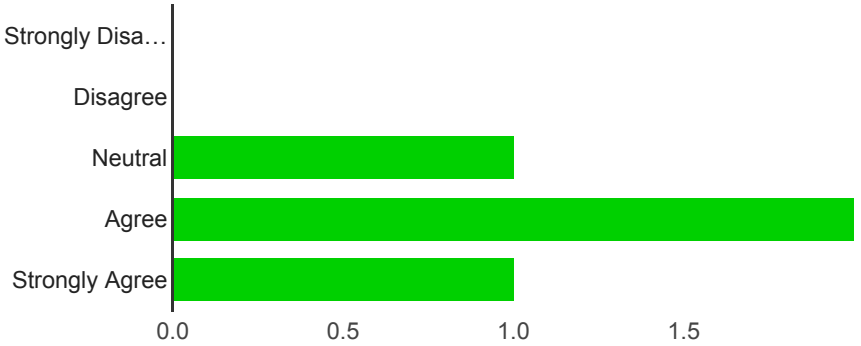
Extended Post-Evaluation Survey

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



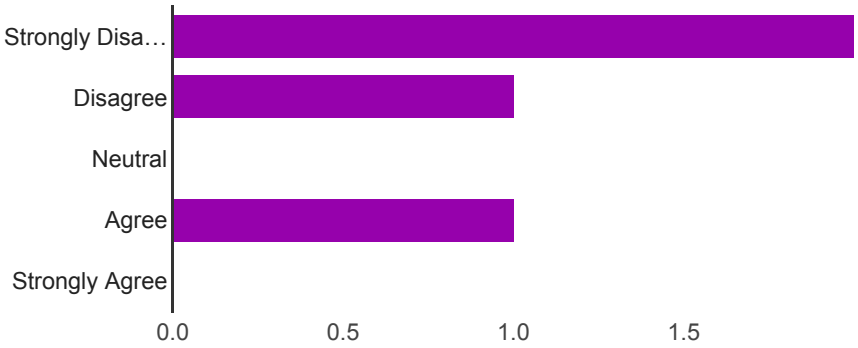
Not very familar:	1	0	0%
	2	0	0%
	3	2	50%
Very familiar:	4	2	50%

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



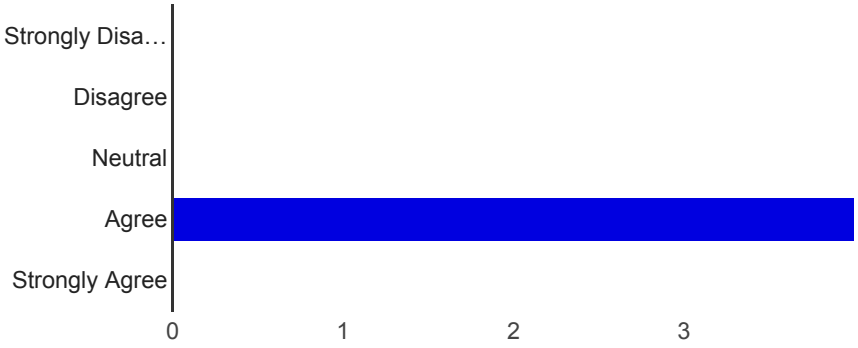
Strongly Disagree	0	0%
Disagree	0	0%
Neutral	1	25%
Agree	2	50%
Strongly Agree	1	25%

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



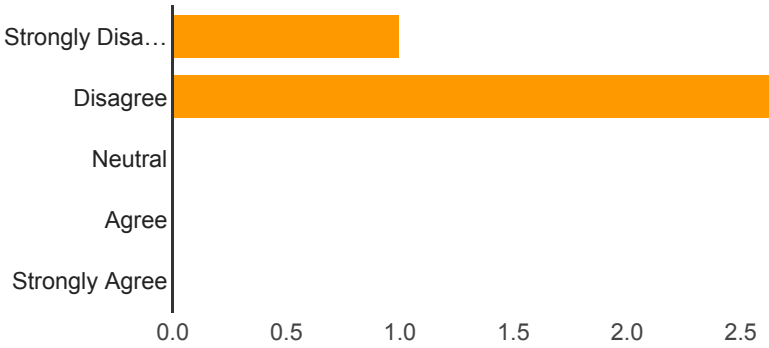
Strongly Disagree	2	50%
Disagree	1	25%
Neutral	0	0%
Agree	1	25%
Strongly Agree	0	0%

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

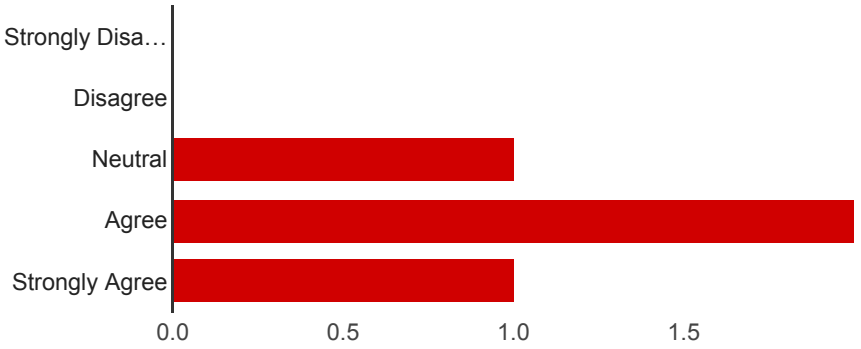


Strongly Disagree	0	0%
Disagree	0	0%
Neutral	0	0%
Agree	4	100%
Strongly Agree	0	0%

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

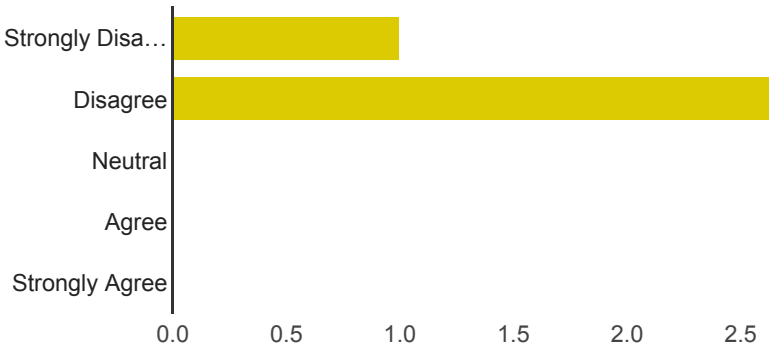


I found the various functions in this system were well integrated. [System Usability Scale]



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

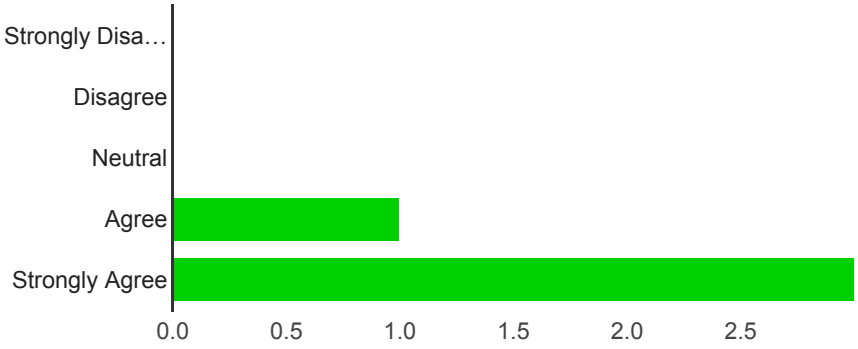
I thought there was too much inconsistency in this system. [System Usability Scale]



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

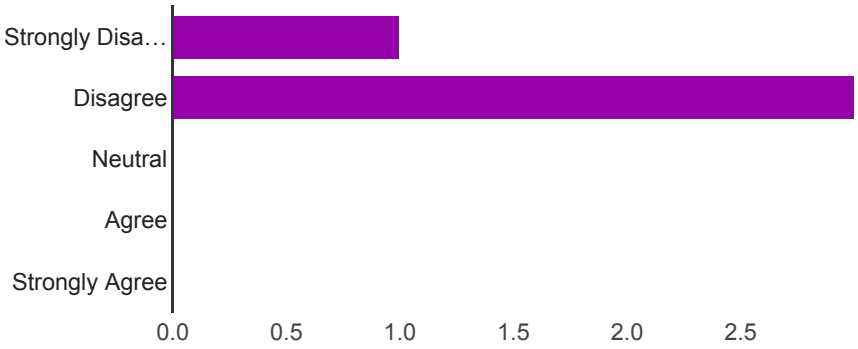
Agree	0	0%
Strongly Agree	0	0%

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



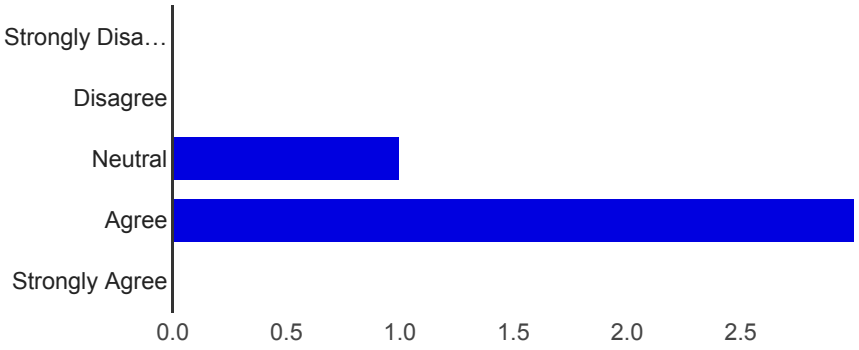
Strongly Disagree	0	0%
Disagree	0	0%
Neutral	0	0%
Agree	1	25%
Strongly Agree	3	75%

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



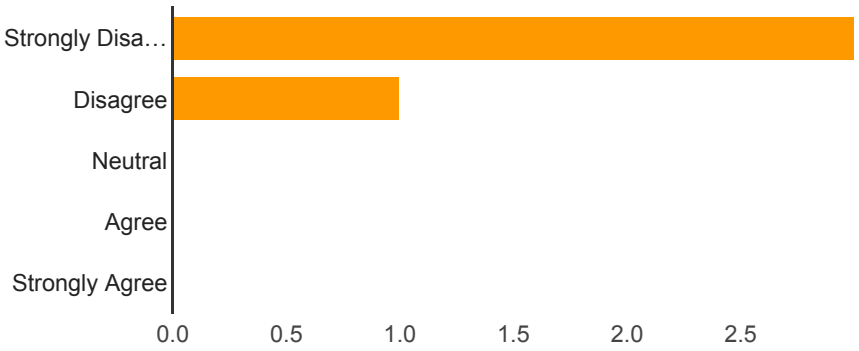
Strongly Disagree	1	25%
Disagree	3	75%
Neutral	0	0%
Agree	0	0%
Strongly Agree	0	0%

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.

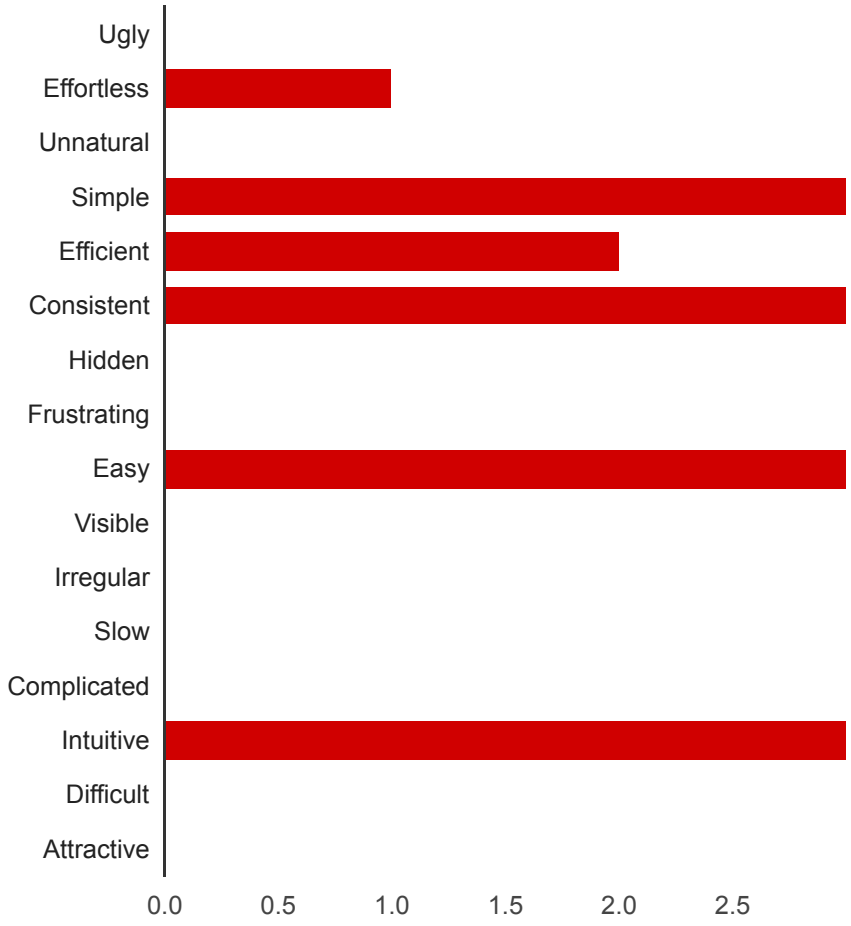
clock in time

graphs were hard to decipher, had to guess were 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

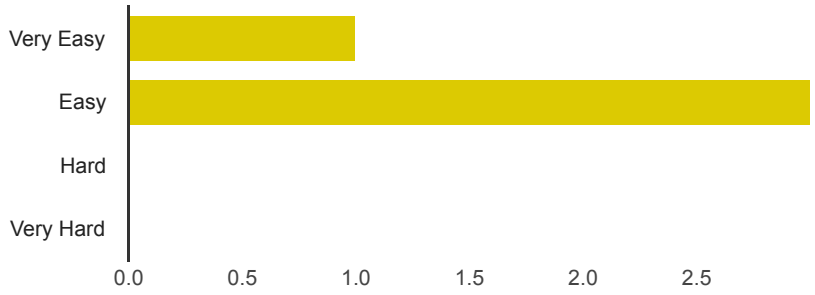
Qualitative Descriptions



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

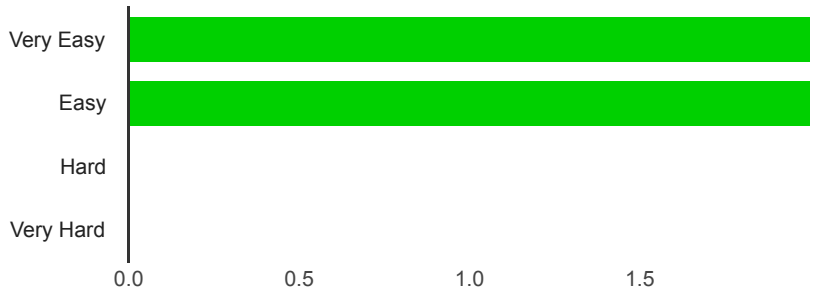
Frustrating	0	0%
Easy	3	75%
Visible	0	0%
Irregular	0	0%
Slow	0	0%
Complicated	0	0%
Intuitive	3	75%
Difficult	0	0%
Attractive	0	0%

Task 1 - Creating Ticket [Task Difficulty]



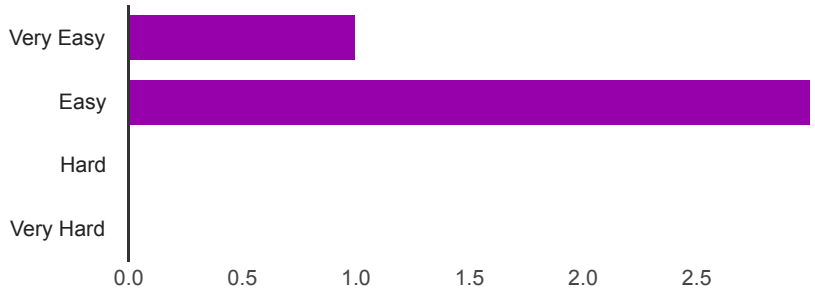
Very Easy	1	25%
Easy	3	75%
Hard	0	0%
Very Hard	0	0%

Task 2 - Closing Ticket [Task Difficulty]



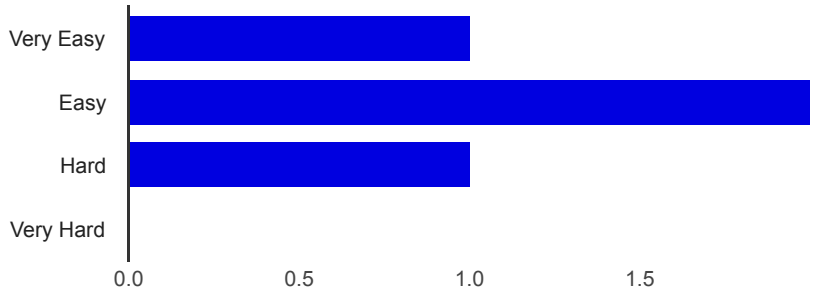
Very Easy	2	50%
Easy	2	50%
Hard	0	0%
Very Hard	0	0%

Task 3 - Clocking On [Task Difficulty]



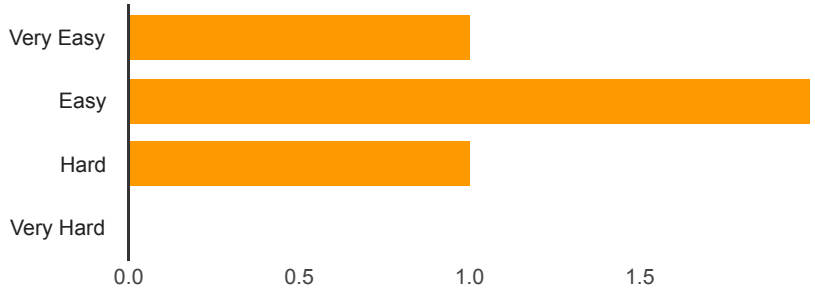
Very Easy	1	25%
Easy	3	75%
Hard	0	0%
Very Hard	0	0%

Task 4 - Resolving Ticket [Task Difficulty]



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

Task 5 - Clocking Off [Task Difficulty]

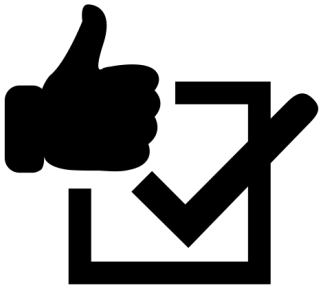


Very Easy	1	25%

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

We follow a Forking workflow[1] when developing Doubtfire.

1.1 Table of Contents

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

[1] See `https://www.atlassian.com/git/tutorials/comparing-workflows/forking-workflow`

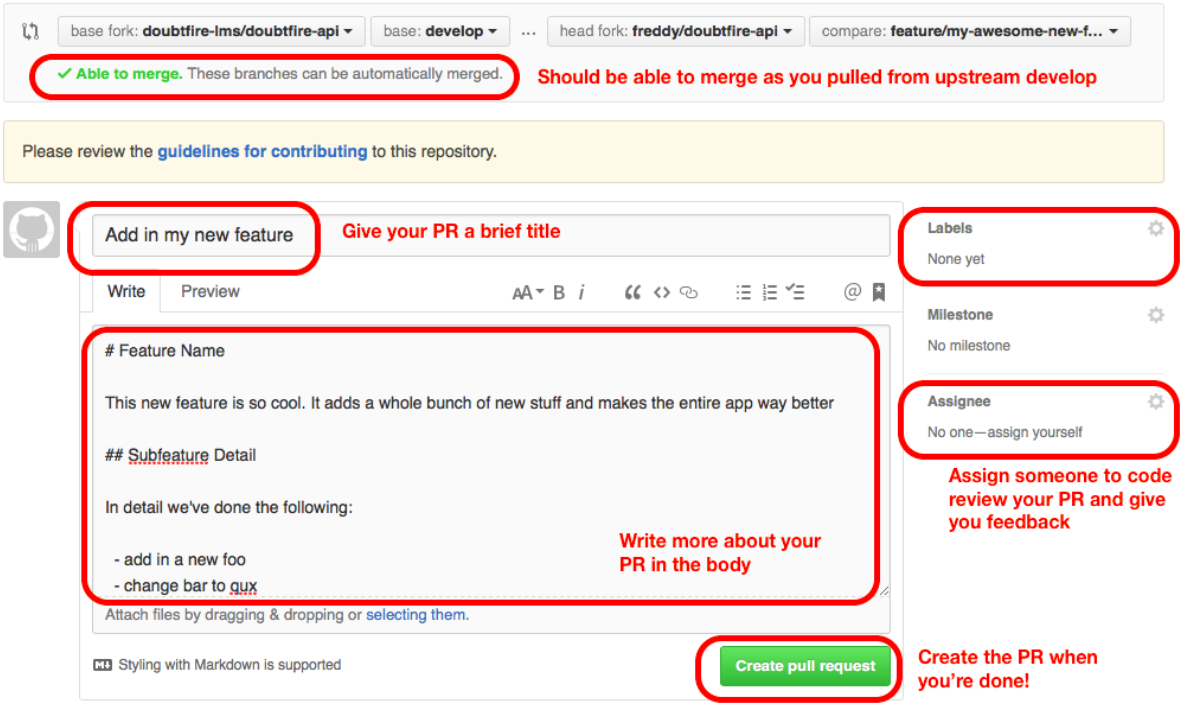


Figure 1: Feature Branches

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*:

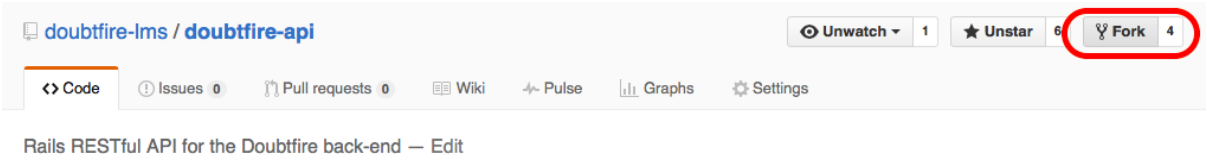


Figure 2: Fork the repo

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:

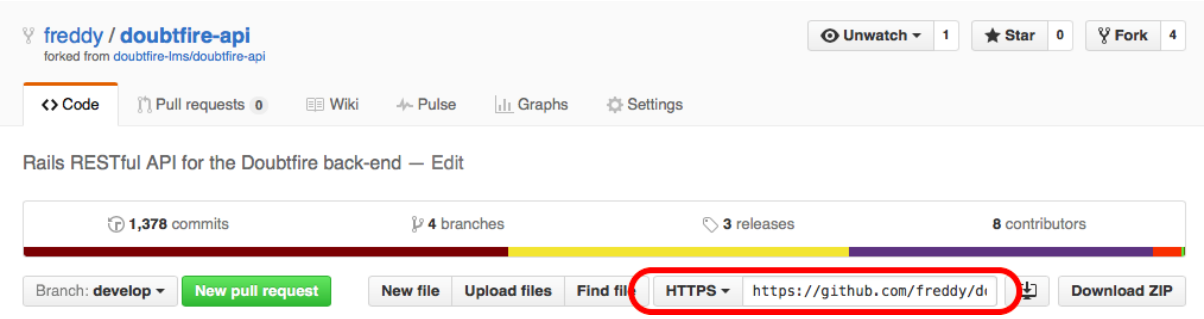


Figure 3: Copy the clone URL

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 `doubfire-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 guide[2] 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
```

[2] See `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":

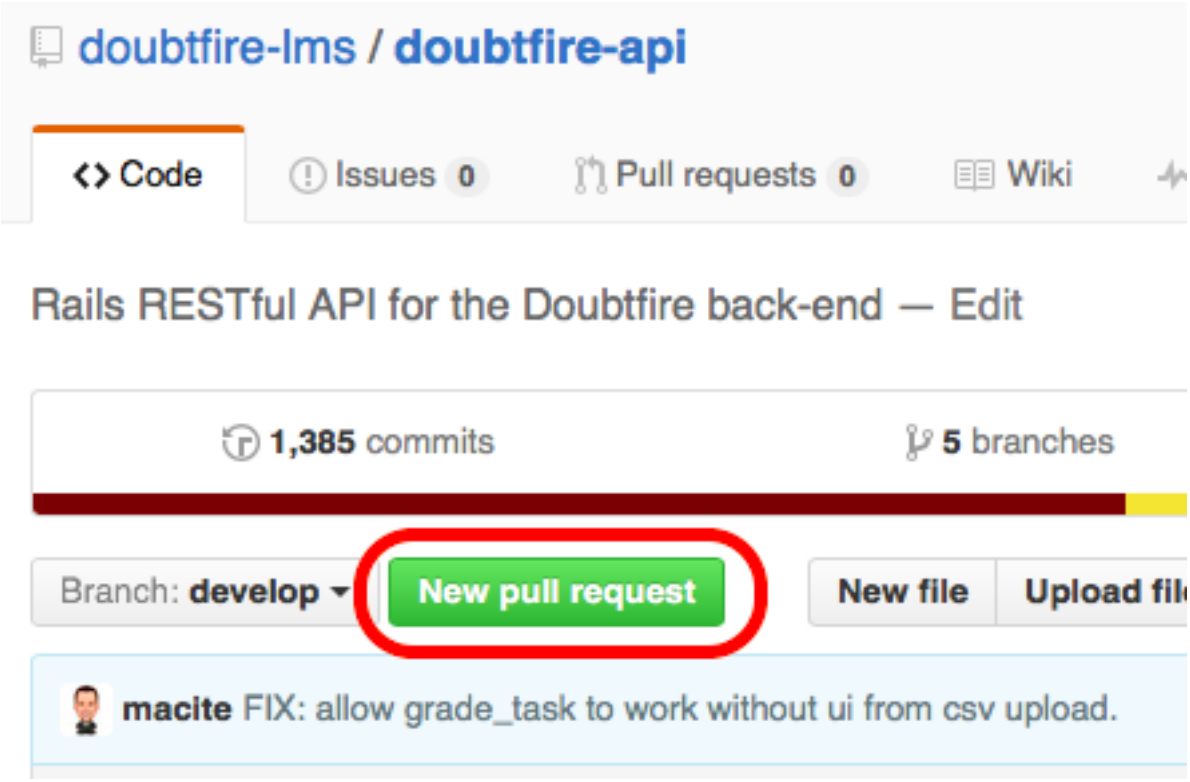


Figure 4: New PR

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.

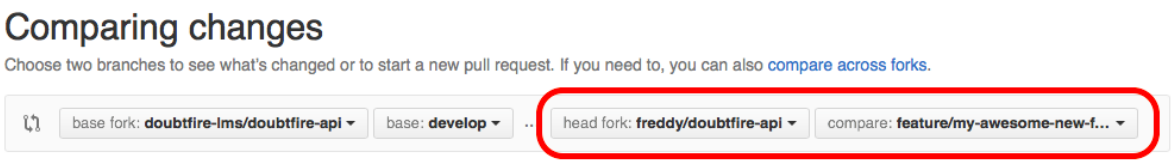


Figure 5: Compare forks

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**.

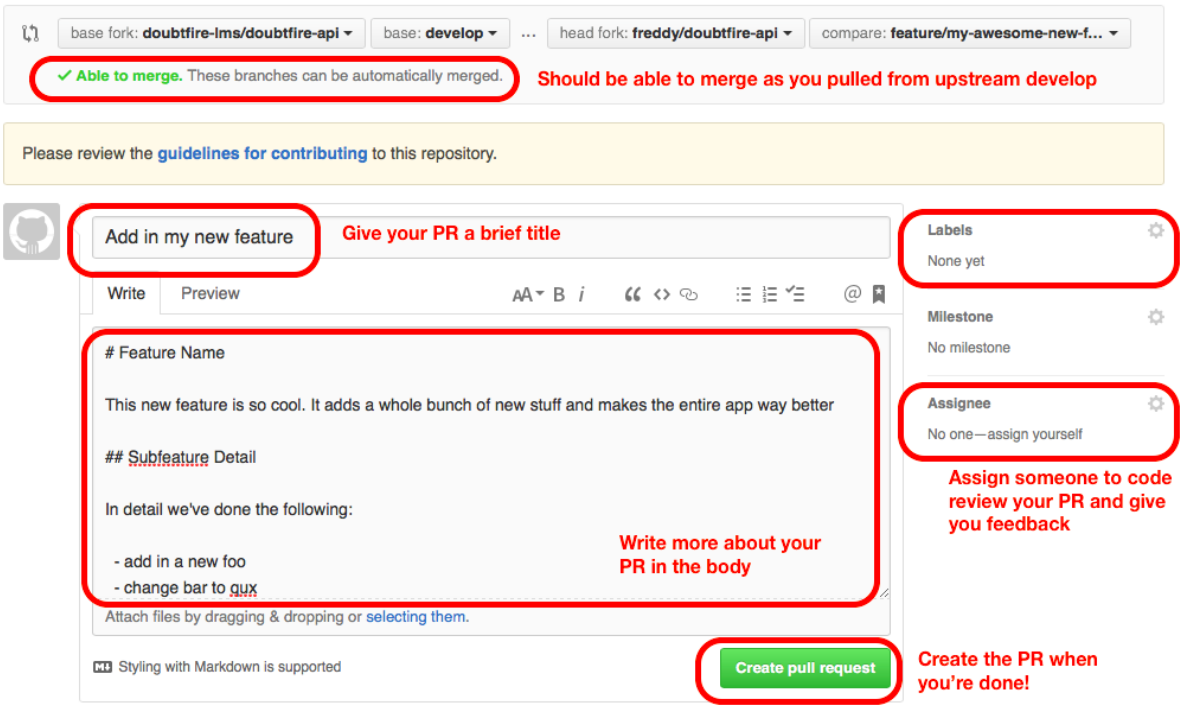


Figure 6: Writing a Pull Request

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 Messages[3].

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
```

[3]See `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 eachother
`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** "Fix*ed* 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 guide[4].

If you are not using the command line for git, you probably should be[5].

2 Contributing To Web

Please read through this document before contributing to Doubtfire.

Before continuing, **please read the contributing document[6] 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 it's role in mind (i.e., as a **Agent Noun**[7]) 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**

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

[5] See `http://try.github.io`

[6] See `https://github.com/doubtfire-lms/doubtfire-api/blob/develop/CONTRIBUTING.md`

[7] See `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)
)
```

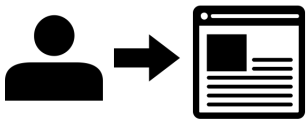
- Always name non-anonymous controllers with a `Ctrl` suffix
- Case correctly:
- `directiveName` should be camelCase - refer to this Angular documentation[8]
- `ServiceName`, `ControllerNameCtrl` should be in PascalCase
- Regardless of abbreviations, stick to these conventions (e.g., `pdfPanelViewer` 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
```

[8] See `https://docs.angularjs.org/guide/directive#normalization`



Chapter 10

User Manual

Contents

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.

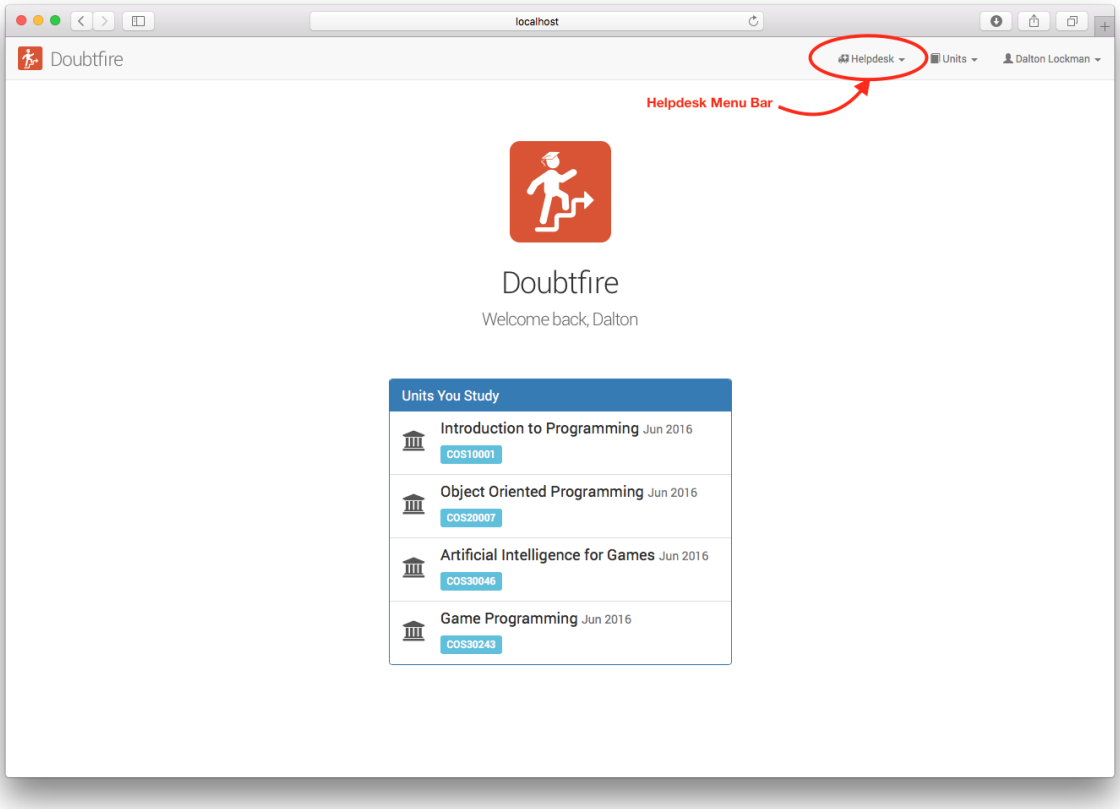


Figure 1: Helpdesk menu

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.

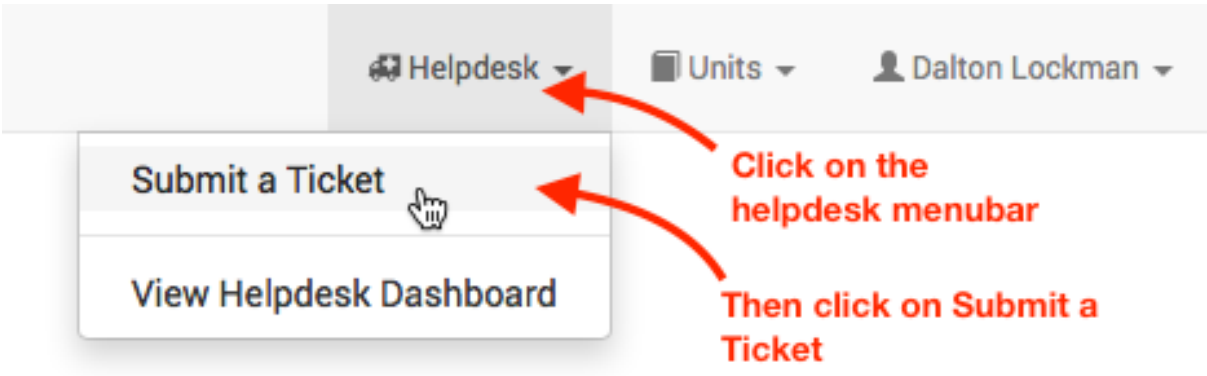
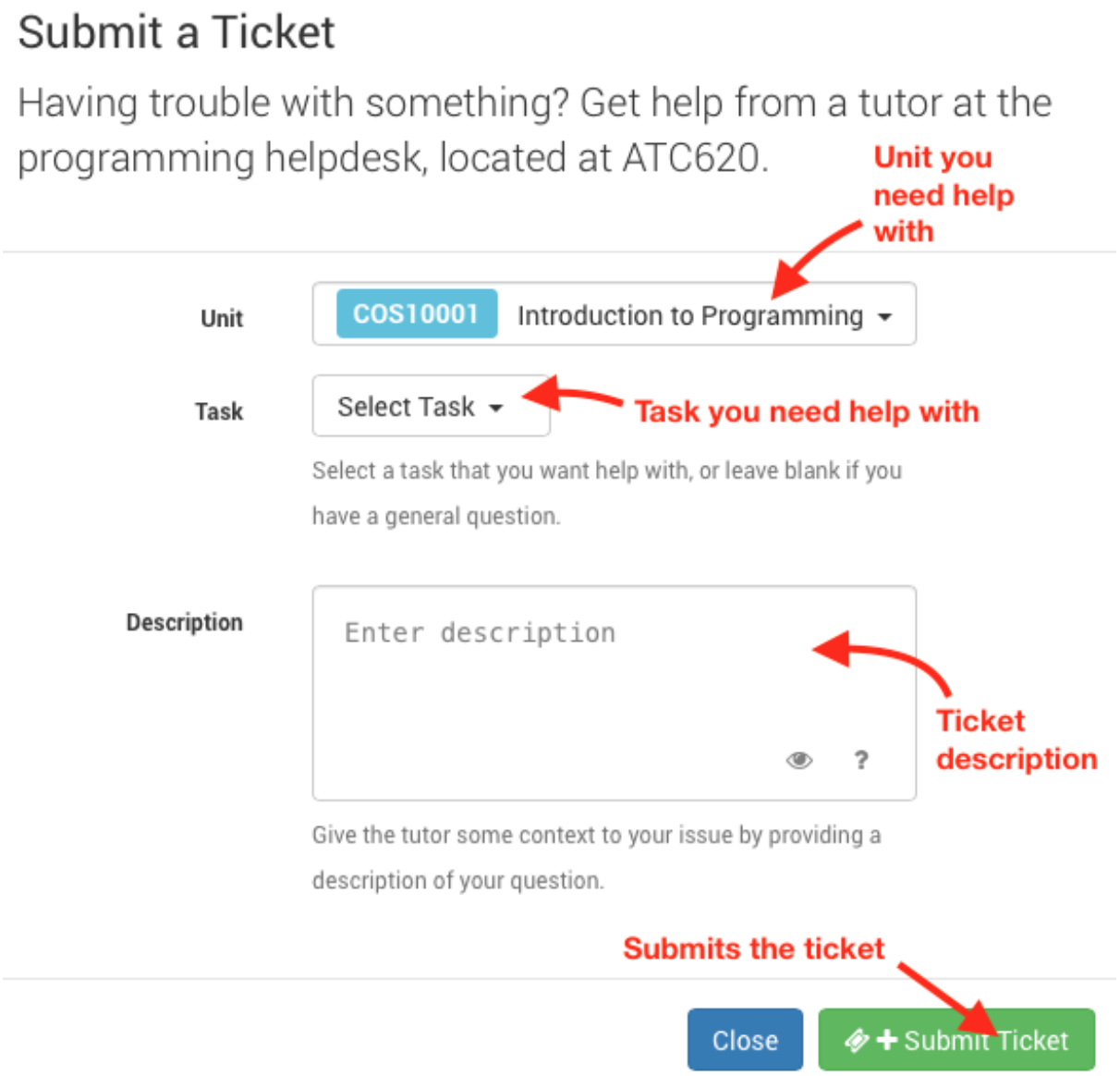


Figure 2: Submit ticket link from the helpdesk menu

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.



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.

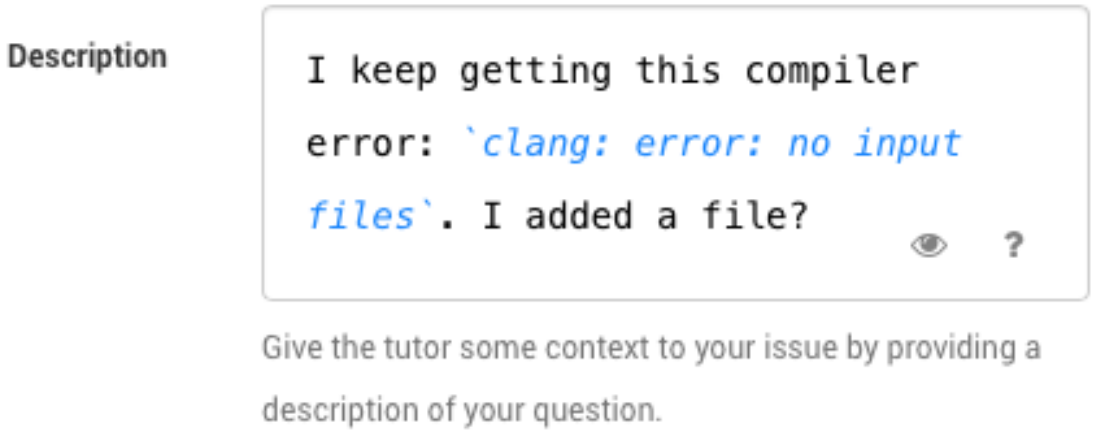


Figure 3: You can add Markdown-formatted code to your description

Once you submit your ticket, as illustrated by the green button in Figure 3, the navigation bar will display an orange ticket sybmol 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.

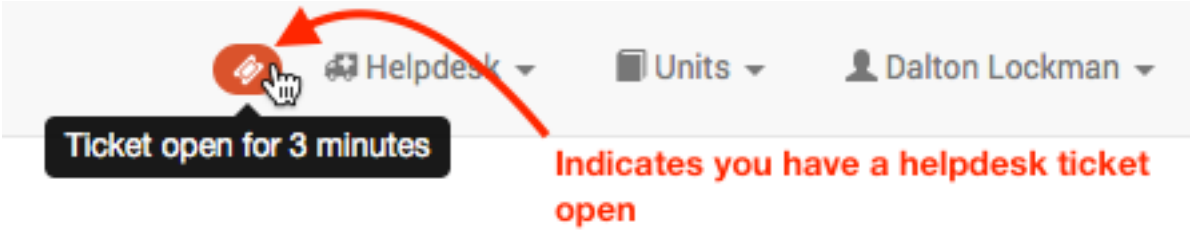


Figure 4: Hovering over a ticket will indicate how long that ticket has been open for

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.

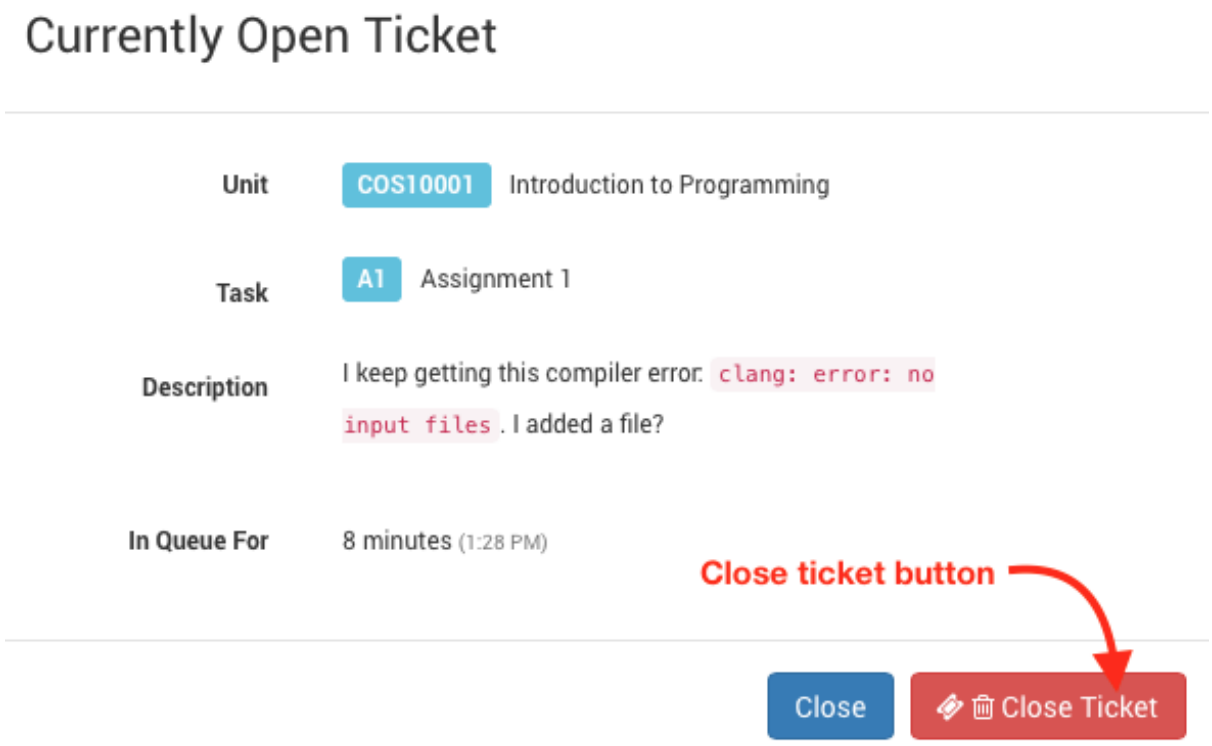


Figure 5: Opening the View Open Ticket link

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.

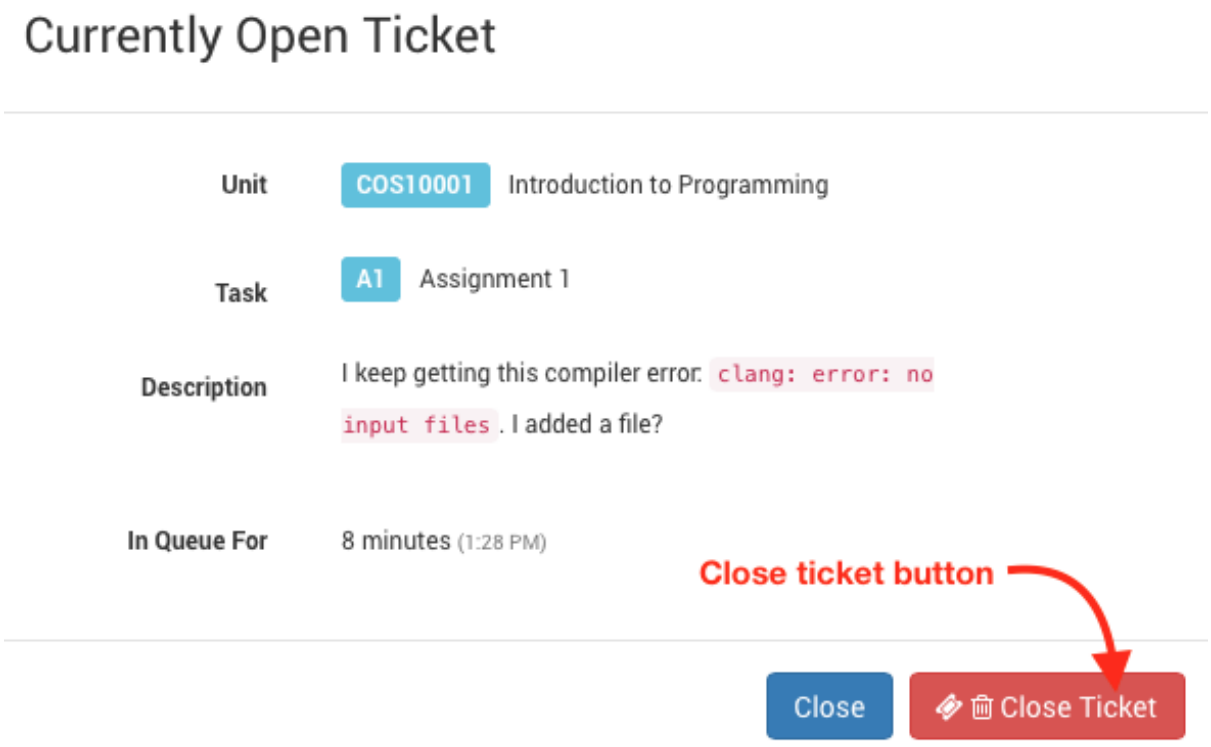


Figure 6: Currently Open Ticket Modal

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.

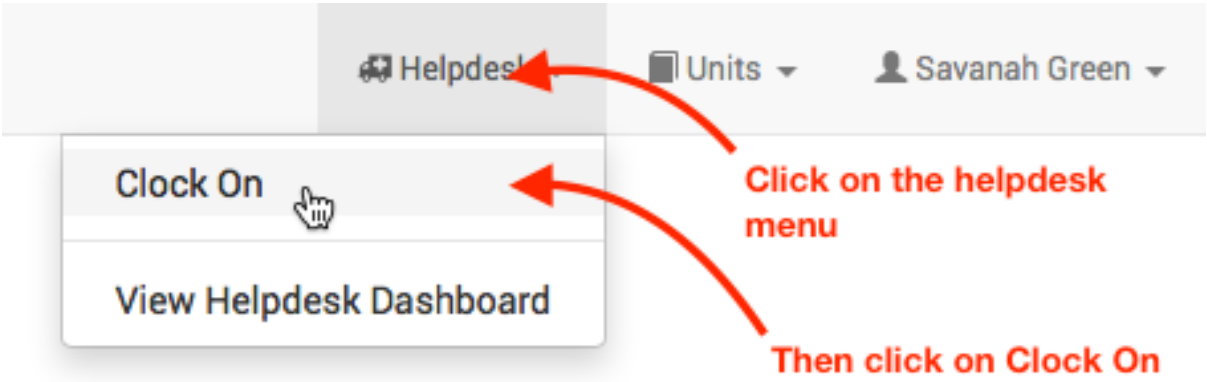


Figure 7: Clock On Link

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.

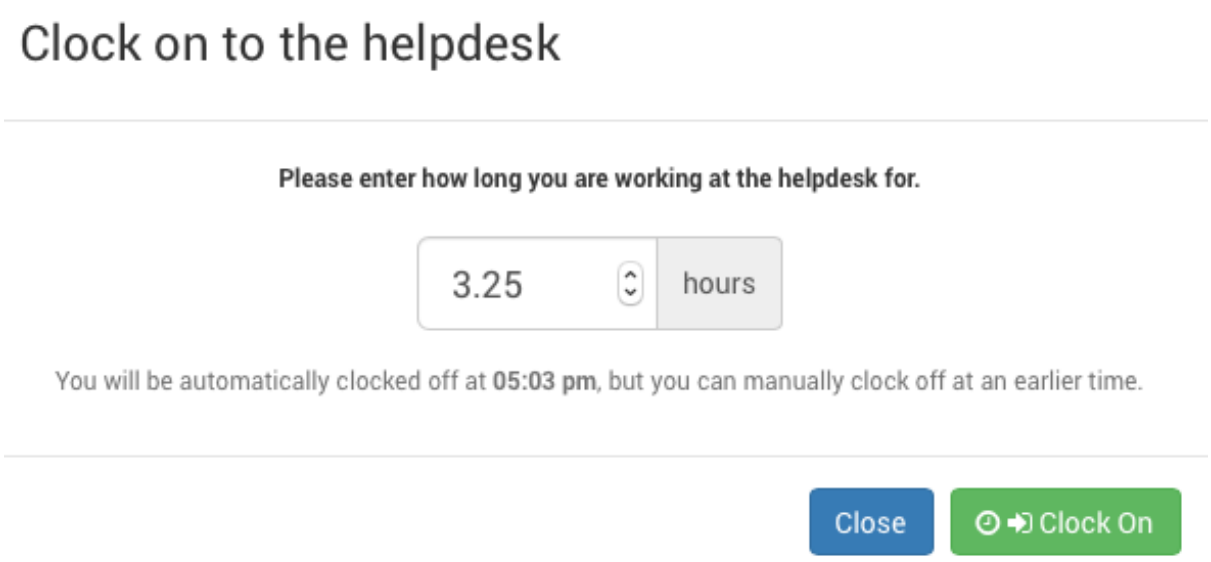


Figure 8: Clocking onto the helpdesk requires you to enter in your work hours

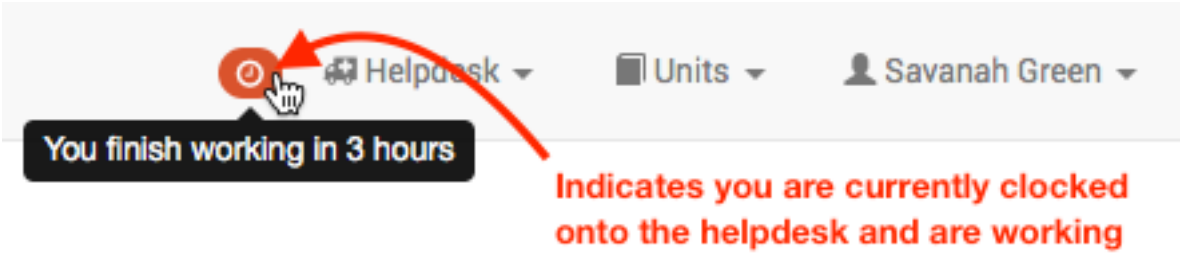
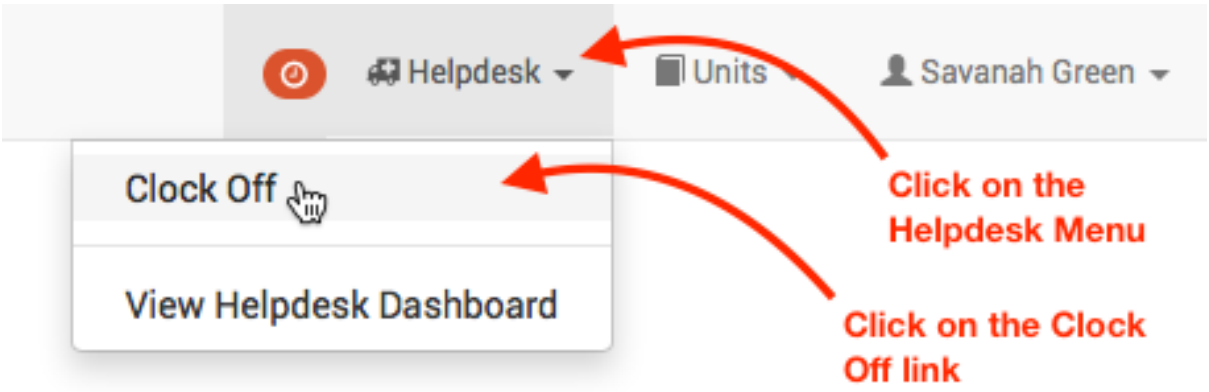


Figure 9: Hovering over the shift icon tells you when the helpdesk will clock you off automatically

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.

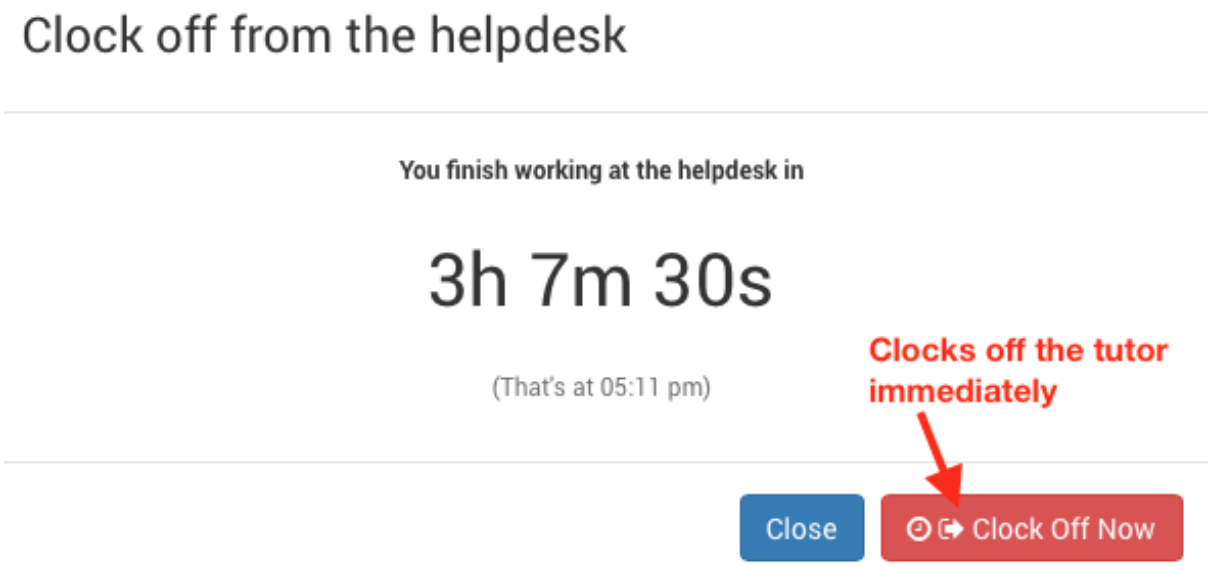
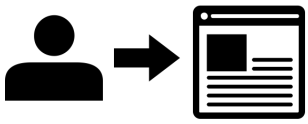


Figure 10: Clock off immediately

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

```
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
```

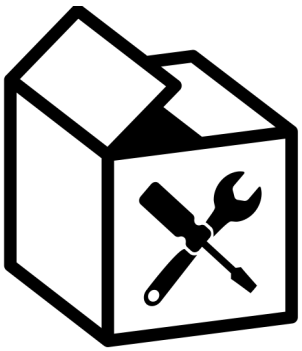
```
|   |   |   \-- user-settings-modal
|   |   |       |-- user-settings-modal.coffee
|   |   |       \-- user-settings-modal.tpl.html
|   |   |-- states
|   |   |   |-- states.coffee
|   |   |   \-- users-admin-view
|   |   |       |-- users-admin-view.coffee
|   |   |       \-- users-admin-view.tpl.html
|   |   \-- 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

See attached document

Commits on Jul 05, 2016

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

Commits on Jul 08, 2016

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

Commits on Jul 11, 2016

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

Commits on Jul 12, 2016

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

Commits on Jul 21, 2016

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

Commits on Jul 22, 2016

alexcu	NEW: Add generator for helpdesk tickets to populator	bb9dde7
alexcu	QUALITY: Remove debugging message from test	9456c4f
alexcu	QUALITY: Move find_or_create_student into own helpder	83b3292
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	0a10dcd
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

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

2 Doubtfire Web Commits

See attached document

Commits on Aug 03, 2016

alexcu	LOOKS: 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	01b6f71
macite	FIX: Ensure task sheet can be viewed	4360588
macite	FIX: Typo in task status text	7849816
lachlanwest	NEW: Add help desk menu item to header menu	e37ea2c
Reubsinit	NEW: Add directory structure for helpdesk ticket modal	6e563fa
Reubsinit	NEW: Add boiler plate code for helpdesk ticket modal	6fb2c9d
lachlanwest	NEW: Basic HTML structure of create ticket form	90bbad5
Reubsinit	NEW: Propagate prototype modal through help desk drop down	92f0d2a
Reubsinit	NEW: Inject projects into scope of ticket modal	3577471
Reubsinit	NEW: Add all units to submit ticket unit dropdown	363ced4
Reubsinit	NEW: Add task dropdown to ticket modal	b10c7aa

Commits on Aug 15, 2016

Reubsinit	NEW: Add HelpDeskTicket to API models	c1ae833
Reubsinit	NEW: Modal now saves ticket to server	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… ⋯	896826a
alexcu	QUALITY: Refactor helpdesk dropdown into directive	ece29e7
alexcu	NEW: Add additional API helpdesk endpoints	dbc7dcc
alexcu	QUALITY: Refactor submit modal	3102231
alexcu	FIX: Ensure query is used for custom state ticket requests	6622d84
alexcu	QUALITY: Replace helpdesk ticket open bool for actual data	26b6f2f
alexcu	ENHANCE: Show pill when ticket is open under header	239e574
alexcu	ENHANCE: Make submit ticket modal into generic ticket modal	585635a
alexcu	ENHANCE: Automatically add/remove ticket icon as needed	fa031d9

Commits on Aug 28, 2016

alexcu	NEW: Add helpdesk session API endpoints	617858d
alexcu	FIX: Fix logic for loading task definition on ticket modal	debd220
alexcu	NEW: Add session modal	2eb903c
alexcu	FIX: Fix right global nav from squishing contents up	3da0ad9
alexcu	ENHANCE: Place alerts underneath navbar	4f12b31

Commits on Aug 29, 2016

alexcu	FIX: Return from current session and ticket funcs on error	b12f363
alexcu	NEW: Add HelpdeskStats factory	e9fcebc
alexcu	QUALITY: Add find project helper method to project service	99d7dc6
alexcu	NEW: Add partially implemented helpdesk-ticket directive	1599cc6
alexcu	NEW: Add partially implemented helpdesk dashboard state	54a923c
alexcu	ENHANCE: Add link to helpdesk state in header dropdown	c52d801
alexcu	QUALITY: Move stats polling into dashboard functionality	65fb6c9
alexcu	FIX: Ensure ticket header data loads only if data is present	717d447
alexcu	QUALITY: Merge data models inside ticket-modal to ngResource	6dfba40
alexcu	FIX: Add missing ngModel attribute to task-def-selector	4f64fca

alexcu	FIX: Ensure polling stops when dashboard controller dies	80341c5

Commits on Sep 10, 2016

alexcu	FIX: Stop flash issue on dashboard for tickets	1bab857
alexcu	NEW: Implement wireframe prototype	fd4ada1
alexcu	ENHANCE: Add position number to enqueued tickets	9dace26
alexcu	FIX: Fix incorrect logic for color warning ranges	78158a3
alexcu	FIX: Ensure time to resolve is rounded	0cc3237
alexcu	QUALITY: Move ticket position to dashboard only	63acb68
alexcu	LOOKS: Centre align tickets on dash	f0ee9b8
alexcu	FIX: Fix breaking create ticket without task	bb72dc3

Commits on Sep 11, 2016

alexcu	NEW: Add graph	400a58d

Commits on Sep 14, 2016

alexcu	FIX: Switch to dashgraph endpoint for graph	b154a6a

Commits on Sep 18, 2016

alexcu	FIX: Fix gruntfile leading comma	cb83dcc
alexcu	LOOKS: Improve headerbar for mobile devices	9af47f2
alexcu	LOOKS: Improve ticket display for mobile devices	a4cda61
alexcu	LOOKS: Show spinner during load of dashboard	4df0412
alexcu	LOOKS: Make buttons in modals clickable on mobile	049cd0a
alexcu	LOOKS: Improve looks for the dashboard menu	42d6b13
alexcu	LOOKS: Hide x overflow for tickets only	e1458dc
alexcu	FIX: Poll on clock on of helpdesk if done via dashboard	ba758d7
alexcu	FIX: Remove overflow on ticket row	eda760c

Commits on Sep 19, 2016

lachlanwest	ENHANCE: Make the dash emoji depend on wait time	1bb198a

Commits on Sep 20, 2016

alexcu	Merge pull request #1 from final-year-project/enhance/status-dependen… ⋯	a004fd4
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
 end

 group :development, :test, :replica do
-  gem 'rspec-rails', '~> 3'
   gem 'factory_girl_rails'
   gem 'minitest-rails'
   gem 'minitest-hyper'
   gem 'database_cleaner'
-  gem "minitest-osx"
+  gem 'minitest-osx'
+  gem 'minitest-around'
 end

 # Student submission
```

22 Gemfile.lock View

```
@@ -81,7 +81,6 @@ GEM
     devise_ldap_authenticatable (0.8.5)
       devise (>= 3.4.1)
       net-ldap (>= 0.6.0, <= 0.11)
-    diff-lcs (1.2.5)
     docile (1.1.5)
     encryptor (1.3.0)
     enumerable-lazy (0.0.1)
@@ -129,6 +128,8 @@ GEM
     mime-types-data (3.2016.0521)
     mini_portile2 (2.1.0)
     minitest (5.9.0)
+    minitest-around (0.3.2)
+      minitest (~> 5.0)
     minitest-hyper (0.2.0)
     minitest-osx (0.1.0)
       minitest (~> 5.4)
@@ -199,23 +200,6 @@ GEM
     responders (2.2.0)
       railties (>= 4.2.0, < 5.1)
     rmagick (2.15.4)
-    rspec-core (3.4.4)
-      rspec-support (~> 3.4.0)
-    rspec-expectations (3.4.0)
-      diff-lcs (>= 1.2.0, < 2.0)
-      rspec-support (~> 3.4.0)
-    rspec-mocks (3.4.1)
-      diff-lcs (>= 1.2.0, < 2.0)
-      rspec-support (~> 3.4.0)
-    rspec-rails (3.4.2)
-      actionpack (>= 3.0, < 4.3)
-      activesupport (>= 3.0, < 4.3)
-      railties (>= 3.0, < 4.3)
-      rspec-core (~> 3.4.0)
-      rspec-expectations (~> 3.4.0)
-      rspec-mocks (~> 3.4.0)
-      rspec-support (~> 3.4.0)
-    rspec-support (3.4.1)
     ruby-filemagic (0.7.1)
     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     -   rspec-rails (~> 3)
291 275   ruby-filemagic
292 276   rubyzip
293 277   simplecov
```

6 app/api/api.rb View

```
    @@ -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 +     mount Api::Helpdesk::Ticket
    36 +     mount Api::Helpdesk::Session
    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 +     AuthHelpers.add_auth_to Api::Helpdesk::Ticket
    58 +     AuthHelpers.add_auth_to Api::Helpdesk::Session
    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

```
... ...  @@ -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 +
```

```
+      # ------------------------------------------------------------------------
+      # GET /helpdesk/sessions?user_id=[id]&is_active=[true|false]
+      # ------------------------------------------------------------------------
+      desc "Get helpdesk sessions"
+      params do
+        optional :user_id,   type: Integer, :desc => "Filter by specific user id"
+        optional :is_active, type: Boolean, :desc => "Filter to only active sessions"
+      end
+      get '/helpdesk/sessions' do
+        unless authorise? current_user, HelpdeskSession, :get_sessions
+          error!({"error" => "Not authorised get helpdesk sessions"}, 403)
+        end
+        if params.empty?
+          logger.info "#{current_user.username} requested all helpdesk sessions"
+          HelpdeskSession.all
+        else
+          user_id = params[:user_id]
+          is_active = params[:is_active]
+          logger.info "#{current_user.username} requested all #{is_active ? 'active' : ''} helpdesk sessions" << (!user
+          sessions = is_active ? HelpdeskSession.active_sessions : HelpdeskSession.all
+          sessions = user_id.nil? ? sessions : sessions.where(user_id: user_id)
+          sessions
+        end
+      end
+
+      # ------------------------------------------------------------------------
+      # POST /helpdesk/sessions
+      # ------------------------------------------------------------------------
+      desc "Begin a new session at the helpdesk as current user"
+      params do
+        requires :clock_off_time, type: DateTime, :desc => "The estimated clock off time when the user ends working"
+        optional :user_id, type: Integer, :desc => "The user who is about to begin working (if blank, posting user is u
+      end
+      post '/helpdesk/sessions' do
+        unless authorise? current_user, HelpdeskSession, :create_session
+          error!({"error" => "Not authorised to create a helpdesk session"}, 403)
+        end
+        user = params[:user_id] ? User.find(params[:user_id]) : current_user
+        # TODO: if current_user != user_id passed in then require PIN
+        if HelpdeskSession.user_clocked_off?(user)
+          session = HelpdeskSession.create!(
+            user: user,
+            clock_on_time: DateTime.now, # Clock on immediately when ticket is created
+            clock_off_time: params[:clock_off_time]
+          )
+          logger.info "#{current_user.username} created new helpdesk session (id=#{session.id})"
+        else
+          error!({"error" => "#{user.username} is already clocked on at the helpdesk"}, 403)
+        end
+        session
+      end
+
+      # ------------------------------------------------------------------------
+      # DELETE /helpdesk/sessions/:id
+      # ------------------------------------------------------------------------
+      desc "Prematurely clock off an existing helpdesk session"
+      params do
+        requires :id, type: Integer, :desc => "The session to clock off"
+      end
+      delete '/helpdesk/sessions/:id' do
+        session = HelpdeskSession.find(params[:id])
+        unless authorise? current_user, session, :clock_off_session
+          error!({"error" => "Not authorised to clock off helpdesk session (id=#{session.id})"}, 403)
```

```
+        end
+        # TODO: if current_user != session.user then require PIN
+        if session.clocked_off?
+          logger.info "#{current_user.username} attempted to clock off already clocked off helpdesk session (id=#{sessi
+        else
+          session.clock_off
+          logger.info "#{current_user.username} prematurely clocked off helpdesk session (id=#{session.id})"
+        end
+        session
+      end
+
+      # ------------------------------------------------------------------------
+      # GET /helpdesk/sessions/tutors
+      # ------------------------------------------------------------------------
+      desc "Get a list of all currently tutors working at the helpdesk"
+      get '/helpdesk/sessions/tutors' do
+        unless authorise? current_user, HelpdeskSession, :get_all_current_session_users
+          error!({"error" => "Not authorised view current helpdesk staff"}, 403)
+        end
+        logger.info "#{current_user.username} requested all currently working helpdesk staff"
+        ActiveModel::ArraySerializer.new(HelpdeskSession.users_working_now, each_serializer: HelpdeskUserSerializer)
+      end
+    end
+  end
+end
```

76 app/api/helpdesk/stats.rb View

```
@@ -0,0 +1,76 @@
+require 'grape'
+
+module Api
+  module Helpdesk
+    #
+    # Helpdesk ticket stats endpoints
+    #
+    class Stats < Grape::API
+      helpers AuthHelpers
+      helpers AuthorisationHelpers
+      helpers LogHelper
+
+      before do
+        authenticated?
+      end
+
+      # ------------------------------------------------------------------------
+      # GET /helpdesk/stats/tickets
+      # ------------------------------------------------------------------------
+      desc 'Gets tickets stats for helpdesk within the duration specified'
+      params do
+        optional :from, type: DateTime, desc: 'The time to start getting statistics (do not provide for all stats)'
+        optional :to,   type: DateTime, desc: 'The time to stop getting statistics (defaults to current time)', default
+      end
+      get '/helpdesk/stats/tickets' do
+        unless authorise? current_user, HelpdeskTicket, :get_ticket_stats
+          error!({error: 'Not authorised to get helpdesk ticket stats'}, 403)
+        end
+
+        logger.info "#{current_user.username} got ticket helpdesk statistics"
+
+        from     = params[:from]
+        to       = params[:to] || DateTime.now
+
```

```
+        {
+          resolved_count:             HelpdeskTicket.resolved_between(from, to).length,
+          number_unresolved:          HelpdeskTicket.all_unresolved.length,
+          average_wait_time_in_mins:  HelpdeskTicket.average_wait_time(from, to)
+        }
+      end
+
+      # ------------------------------------------------------------------------
+      # GET /helpdesk/stats/dashgraph
+      # ------------------------------------------------------------------------
+      desc 'Gets dashboard graph data for the helpdesk (last 3 hours)'
+      get '/helpdesk/stats/dashgraph' do
+        unless authorise? current_user, HelpdeskTicket, :get_ticket_stats
+          error!({error: 'Not authorised to get helpdesk ticket stats'}, 403)
+        end
+
+        logger.info "#{current_user.username} got ticket helpdesk dash graph data"
+
+        # Work from now downto 3 hours ago
+        graph_range = 3.hours.ago.to_i..DateTime.now.to_i
+        graph_data = {}
+        # Collect data every minute going backwards -- stats for every min in
+        # last 180 mins
+        interval = 1.minute.to_i
+        graph_range.first.step(graph_range.last, interval) do |unix_t|
+          # Range is the time between graph_time less time mins ago
+          # to graph_time
+          range = Time.at(unix_t + interval).utc..Time.at(unix_t).utc
+          # Stats to include
+          avg_wait   = HelpdeskTicket.average_wait_time(range.first, range.last)
+          unresolved = HelpdeskTicket.unresolved_between(range.first, range.last)
+          # Insert at this time the stats there were
+          graph_data[unix_t] = {
+            average_wait_time_in_mins:    avg_wait,
+            number_of_unresolved_tickets: unresolved.length # count only
+          }
+        end
+        graph_data
+      end
+    end
+  end
+end
```

120 app/api/helpdesk/ticket.rb View

```
@@ -0,0 +1,120 @@
+require 'grape'
+require 'helpdesk_ticket_serializer'
+
+module Api
+  module Helpdesk
+    #
+    # Helpdesk ticketing system endpoints
+    #
+    class Ticket < Grape::API
+      helpers AuthHelpers
+      helpers AuthorisationHelpers
+      helpers LogHelper
+
+      before do
+        authenticated?
+      end
+
```

```
+      # ----------------------------------------------------------------------
+      # POST /helpdesk/tickets
+      # ----------------------------------------------------------------------
+      desc "Add a new helpdesk ticket"
+      params do
+        requires :project_id, type: Integer, desc: "The project to assign the ticket to"
+        optional :task_definition_id, type: Integer, desc: "Task which the student needs help with"
+        optional :description, type: String, desc: "Description associated to the ticket"
+      end
+      post '/helpdesk/tickets' do
+        project = Project.find(params[:project_id])
+        task = params[:task_definition_id].nil? ? nil : task = project.task_for_task_definition(params[:task_definition
+
+        unless authorise? current_user, project, :create_ticket
+          error!({error: "Not authorised to create a ticket for project #{project.id}"}, 403)
+        end
+
+        # Only allow them to create a new ticket if they haven't done so already
+        if HelpdeskTicket.user_has_ticket_open? project.user.id
+          logger.info "#{current_user.username} tried to create new ticket but already has one open"
+          error!({error: "User already has a ticket open"}, 403)
+        end
+
+        ticket = HelpdeskTicket.create!({
+          project: project,
+          task: task,
+          description: params[:description]
+        })
+
+        logger.info "#{current_user.username} created new ticket #{ticket.id} for #{ticket.unit.name}"
+        ticket
+      end
+
+      # ----------------------------------------------------------------------
+      # GET /helpdesk/tickets?filter={resolved,unresolved,closed,all}&user_id={id}
+      # Optional User Id and Filter. Filter default to all.
+      # ----------------------------------------------------------------------
+      desc "Gets all helpdesk tickets. Optional User Id and Resolved filter."
+      params do
+        optional :user_id, type: String, desc: "The id of the user to get tickets for."
+        optional :filter, type: String, desc: "Filter by resolved, unresolved, closed or all. Defaults to all.", defaul
+        optional :shallow, type: Boolean, desc: "Use shallow serializer vs detailed serializer", default: true
+      end
+      get '/helpdesk/tickets' do
+        unless authorise? current_user, HelpdeskTicket, :get_tickets
+          error!({error: 'Not authorised to get tickets'}, 403)
+        end
+
+        user_id = params[:user_id]
+        filter = params[:filter] || 'all'
+        tickets = HelpdeskTicket.all_by_state(filter.to_sym, user_id)
+
+        serializer = params[:shallow] ? ShallowHelpdeskTicketSerializer : HelpdeskTicketSerializer
+        ActiveModel::ArraySerializer.new(tickets, each_serializer: serializer)
+      end
+
+      # ----------------------------------------------------------------------
+      # GET /helpdesk/tickets/:id
+      # ----------------------------------------------------------------------
+      desc "Gets helpdesk ticket with an id"
+      params do
+        requires :id, type: Integer, desc: "The id of the ticket to get"
+      end
```

```
+      get '/helpdesk/tickets/:id' do
+        ticket = HelpdeskTicket.find(params[:id])
+
+        if not authorise? current_user, ticket, :get_details
+          error!({error: 'Not authorised to get ticket details'}, 403)
+        end
+
+        ticket
+      end
+
+      # ------------------------------------------------------------------------
+      # DELETE /helpdesk/tickets/:id?[resolve=true|false]
+      # ------------------------------------------------------------------------
+      desc "Updates helpdesk ticket with an id"
+      params do
+        requires :id, type: Integer, desc: "The id to of the ticket to delete"
+        optional :resolve, type: Boolean, desc: "Mark the ticket as resolved"
+      end
+      delete '/helpdesk/tickets/:id' do
+        ticket = HelpdeskTicket.find(params[:id])
+
+        if params[:resolve] == true
+          unless authorise? current_user, ticket, :resolve_ticket
+            error!({error: "Not authorised to resolve ticket #{params[:id]}"}, 403)
+          end
+          ticket.resolve
+          logger.info "#{current_user.username} resolved ticket #{ticket.id}"
+        else
+          unless authorise? current_user, ticket, :close_ticket
+            error!({error: "Not authorised to close ticket #{params[:id]}"}, 403)
+          end
+          ticket.close
+          logger.info "#{current_user.username} closed ticket #{ticket.id}"
+        end
+
+        ticket
+      end
+    end
+  end
+end
```

179 app/models/helpdesk_session.rb View

```
@@ -0,0 +1,179 @@
+#
+# Tracking for sessions of staff working at the helpdesk.
+#
+class HelpdeskSession < ActiveRecord::Base
+  # Model associations
+  belongs_to :user
+
+  # Model constratins/validation
+  validates :user, presence: true
+  validates :clock_on_time, presence: true
+  validates :clock_off_time, presence: true
+
+  # Validates that a clock off time is always after the clock on time
+  validates_presence_of :clock_on_time, if: :clock_off_must_be_after_clock_on
+  def clock_off_must_be_after_clock_on
+    if clock_off_time && clock_off_time < clock_on_time
+      errors.add(:clock_off_time, "can't be before :clock_on_time")
+    end
+  end
```

```
+
+  # Validates that the user for a session is not a student
+  validates_presence_of :user, if: :user_is_not_student
+  def user_is_not_student
+    if user && user.role == Role.student
+      errors.add(:user, 'must not be a student')
+    end
+  end
+
+  #
+  # Permissions around helpdesk sessions
+  #
+  def self.permissions
+    # What can students do with sessions?
+    student_role_permissions = [
+      :get_all_current_session_users
+    ]
+    # What can tutors do with sessions?
+    tutor_role_permissions = [
+      :create_session,
+      :clock_off_session,
+      :get_all_current_session_users,
+      :get_sessions
+    ]
+    # What can convenors do with sessions?
+    convenor_role_permissions = [
+      :create_session,
+      :clock_off_session,
+      :get_all_current_session_users,
+      :get_sessions,
+      :get_ticket_stats
+    ]
+    # What can admins do with sessions?
+    admin_role_permissions = [
+      :create_session,
+      :clock_off_session,
+      :get_all_current_session_users,
+      :get_sessions,
+      :get_ticket_stats
+    ]
+    # What can nil users do with sessions?
+    nil_role_permissions = [
+    ]
+
+    # Return permissions hash
+    {
+      :admin    => admin_role_permissions,
+      :convenor => convenor_role_permissions,
+      :tutor    => tutor_role_permissions,
+      :student  => student_role_permissions,
+      :nil      => nil_role_permissions
+    }
+  end
+
+  def self.role_for(user)
+    user.role
+  end
+
+  def role_for(user)
+    # If the user provided is the user registered to the session, then
+    # they can do whatever they want with it. Convenors and admins can
+    # override this
+    if user == self.user || [Role.convenor, Role.admin].include?(user.role)
```

```
+      user.role
+    end
+  end
+
+  #
+  # Override the clock off time to set it to now
+  #
+  def clock_off
+    self.clock_off_time = DateTime.now
+    save!
+  end
+
+  #
+  # Returns the staff sessions within the given timeframe grouped by user id
+  #
+  def self.sessions_by_staff_between(from = nil, to = DateTime.now)
+    sessions_between(from, to).group_by(&:user)
+  end
+  def self.sessions_by_staff
+    sessions_by_staff_between
+  end
+
+  #
+  # Returns all session stats between the timeframe
+  #
+  def self.stats_by_staff_id(from = nil, to = DateTime.now)
+    sessions_by_staff_between(from, to).map do |user, sessions|
+      durations = sessions.map(&:session_duration)
+      average_duration = durations.inject(:+) / durations.length
+      count = sessions.length
+      {
+        user_id: user.id,
+        average_duration_in_hours: average_duration,
+        count: count
+      }
+    end
+  end
+
+  #
+  # Returns all sessions within the time-frame given
+  #
+  def self.sessions_between(from = nil, to = DateTime.now)
+    to ||= DateTime.now
+    return where('clock_on_time > ? AND clock_off_time < ?', from, to) unless from.nil?
+    where('clock_off_time < ?', to)
+  end
+
+  #
+  # Returns all currently active sessions
+  #
+  def self.active_sessions
+    where('clock_off_time > ?', DateTime.now)
+  end
+
+  #
+  # Returns all users currently working now
+  #
+  def self.users_working_now
+    active_sessions.map(&:user)
+  end
+
+  #
+  # Returns true if this session is clocked off
```

```
+  #
+  def clocked_off?
+    clock_off_time < DateTime.now
+  end
+
+  #
+  # Returns true if this session is clocked on
+  #
+  def clocked_on?
+    !clocked_off?
+  end
+
+  #
+  # Checks if the user provided is currently clocked off
+  #
+  def self.user_clocked_off?(user)
+    # Should have no sessions clocked on thus nothing found clocked on
+    user.helpdesk_sessions.find(&:clocked_on?).nil?
+  end
+
+  #
+  # Checks if the user provided is currently clocked on
+  #
+  def self.user_clocked_on?(user)
+    !user_clocked_off(user)
+  end
+
+  #
+  # Returns the duration of time for a session in hours
+  #
+  def session_duration
+    (clock_off_time - clock_on_time) / 60 / 60
+  end
+end
```

170 app/models/helpdesk_ticket.rb View

```
...  @@ -0,0 +1,170 @@
1    +#
2    +# Tracking for tickets logged at the helpdesk.
3    +#
4    +class HelpdeskTicket < ActiveRecord::Base
5    +  # Model associations
6    +  belongs_to :project
7    +  belongs_to :task
8    +
9    +  # Model constratins/validation
10   +  validates :project, presence: true # Must always be associated to a project
11   +
12   +  #
13   +  # Permissions around group data
14   +  #
15   +  def self.permissions
16   +    # What can students do with all tickets?
17   +    student_role_permissions = [
18   +      :get_details,
19   +      :get_tickets,
20   +      :get_ticket_stats,
21   +      :close_ticket
22   +    ]
23   +    # What can tutors|convenors|admins do with all tickets?
24   +    tutor_role_permissions = convenor_role_permissions = admin_role_permissions = [
25   +      :get_tickets,
```

```
+      :get_details,
+      :get_ticket_stats,
+      :close_ticket,
+      :resolve_ticket
+    ]
+    # What can nil users do with all tickets?
+    nil_role_permissions = [
+
+    ]
+
+    # Return permissions hash
+    {
+      :admin    => admin_role_permissions,
+      :convenor => convenor_role_permissions,
+      :tutor    => tutor_role_permissions,
+      :student  => student_role_permissions,
+      :nil      => nil_role_permissions
+    }
+  end
+
+  def self.role_for(user)
+    user.role
+  end
+
+  def role_for(user)
+    if user == project.user
+      Role.student
+    elsif user.role != Role.student
+      user.role
+    else
+      nil
+    end
+  end
+
+  # Returns true if a used has a ticket open
+  def self.user_has_ticket_open?(user_id)
+    !all_unresolved(user_id).empty?
+  end
+
+  # Returns back all unresolved tickets, optionally limit to a user
+  def self.all_unresolved(user_id = nil)
+    all_by_state(:unresolved, user_id)
+  end
+
+  # Returns back all resolved tickets, optionally limit to a user
+  def self.all_resolved(user_id = nil)
+    all_by_state(:resolved, user_id)
+  end
+
+  # Returns back all closed tickets, optionally limit to a user
+  def self.all_closed(user_id = nil)
+    all_by_state(:closed, user_id)
+  end
+
+  # Finds tickets of a particular status, optionally limit to a user
+  def self.all_by_state(resolved_filter, user_id = nil)
+    tickets =
+      case resolved_filter
+      when :resolved
+        where(is_resolved: true)
+      when :unresolved
+        where(is_closed: false)
+      when :closed
```

```
+        where(is_closed: true, is_resolved: false)
+      else
+        all
+      end
+
+    unless user_id.nil?
+      user = User.find(user_id)
+      project = Project.for_user(user, false) # see app/models/project.rb:76
+      tickets = tickets.where(project: project) # limits the scope of tickets down to those with the project provided
+    end
+
+    tickets.order(:created_at)
+  end
+
+  # Get all tickets resolved between two dates
+  def self.resolved_between(from = nil, to = DateTime.now)
+    to ||= DateTime.now # if nil is passed in
+    from ? all_resolved.where(closed_at: from..to) : all_resolved
+  end
+
+  #
+  # Get all tickets that were still unresolved between from and to
+  # dates
+  #
+  def self.unresolved_between(from, to = DateTime.now)
+    to ||= DateTime.now # if nil passed in
+    where('created_at <= ? AND (closed_at >= ? OR NOT is_closed)', from, to)
+  end
+
+  # Calculates the average time to resolve a ticket from the duration
+  # provided (i.e., between from and now). If no arguments are provided,
+  # all resolved tickets will be used regardless of when they were resolved.
+  # Where no tickets are found within the period, nil is returned.
+  def self.average_resolve_time_between(from = nil, to = DateTime.now)
+    tickets = resolved_between(from, to)
+    resolved_between(from, to).average(:minutes_to_resolve).to_i unless tickets.empty?
+  end
+
+  #
+  # Determines the current average wait time for a ticket
+  #
+  def self.average_wait_time(from, to)
+    avg_resolve = HelpdeskTicket.average_resolve_time_between(from, to) || 0
+    num_tickets = HelpdeskTicket.unresolved_between(from, to).length
+    avg_resolve * num_tickets
+  end
+
+  # Resolves the ticket
+  def resolve
+    unless is_closed
+      # Resolving closes the ticket
+      close
+      self.is_resolved = true
+      self.minutes_to_resolve = ((closed_at - created_at) / 60).to_f.round(2)
+      save!
+    end
+  end
+
+  # Unit for ticket
+  def unit
+    project.unit
+  end
+
```

```
+  # Student for ticket
+  def student
+    project.student
+  end
+
+  # Returns true if ticket is associated with a task
+  def task?
+    !task.nil?
+  end
+
+  # Prematurely closes a ticket
+  def close
+    unless is_closed
+      self.is_closed = true
+      self.closed_at = DateTime.now
+      save!
+    end
+  end
+end
```

7 app/models/project.rb View

```
@@ -21,6 +21,8 @@ class Project < ActiveRecord::Base

   has_many :learning_outcome_task_links, through: :tasks

+  has_many :helpdesk_tickets, dependent: :destroy
+
   validate :must_be_in_group_tutorials

   #
@@ -33,7 +35,8 @@ def self.permissions
       :change_tutorial,
       :make_submission,
       :get_submission,
-      :change
+      :change,
+      :create_ticket
     ]
     # What can tutors do with projects?
     tutor_role_permissions = [
@@ -860,6 +863,8 @@ def has_task_for_task_definition?(td)
   # task if the task does not exist for this project.
   #
   def task_for_task_definition(td)
+    # Find the definition if a ID is passed in.
+    td = TaskDefinition.find(td) if td.is_a? Fixnum
     logger.debug "Finding task #{td.abbreviation} for project #{log_details()}"
     result = tasks.where(task_definition: td).first
     if result.nil?
```

1 app/models/task.rb View

```
@@ -74,6 +74,7 @@ def role_for(user)
   has_many :reverse_plagiarism_match_links, class_name: "PlagiarismMatchLink", dependent: :destroy, inverse_of: :other_
   has_many :learning_outcome_task_links, dependent: :destroy # links to learning outcomes
   has_many :learning_outcomes,  through: :learning_outcome_task_links
+  has_many :helpdesk_tickets

   validates :task_definition_id, uniqueness: { scope: :project,
     message: "must be unique within the project" }
```

5 app/models/user.rb View

```
@@ -67,6 +67,7 @@ def reset_authentication_token!
   belongs_to  :role   # Foreign Key
   has_many    :unit_roles, dependent: :destroy
   has_many    :projects
+  has_many    :helpdesk_sessions

   # Model validations/constraints
   validates :first_name,  presence: true
@@ -407,4 +408,8 @@ def self.import_from_csv(current_user, file)
       errors:  errors
     }
   end
+
+  def units_taught
+    unit_roles.map(&:unit)
+  end
 end
```

15 app/serializers/helpdesk_session_serializer.rb View

```
@@ -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
```

40 app/serializers/helpdesk_ticket_serializer.rb View

```
@@ -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
```

```
+    task = object.task
+    ShallowTaskDefinitionSerializer.new task.task_definition if task
+  end
+end
+
+class ShallowHelpdeskTicketSerializer < ActiveModel::Serializer
+  attributes :id,
+             :project_id,
+             :task_definition_id,
+             :description,
+             :is_resolved,
+             :is_closed,
+             :created_at,
+             :closed_at,
+             :minutes_to_resolve
+
+  def task_definition_id
+    task = object.task
+    task.task_definition_id if task
+  end
+end
```

25 app/serializers/project_serializer.rb View

```
@@ -10,6 +10,31 @@ class ShallowProjectSerializer < ActiveModel::Serializer
   def project_id
     object.id
   end
+
+  def student_name
+    object.student.name
+  end
+
+  def unit_id
+    object.unit.id
+  end
+
+  def unit_code
+    object.unit.code
+  end
+
+  def unit_name
+    object.unit.name
+  end
+
+  def tutor_name
+    tutorial = object.tutorial
+    tutorial.tutor.name if tutorial
+  end
+
+  def start_date
+    object.unit.start_date.to_date
+  end
 end

 # Student project serializer is used with teaching staff
```

4 app/serializers/task_definition_serializer.rb View

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

```
+end
+
 class TaskDefinitionSerializer < ActiveModel::Serializer
   attributes :id, :abbreviation, :name, :description,
     :weight, :target_grade, :target_date,
```

8 app/serializers/user_serializer.rb View

```
@@ -10,3 +10,11 @@ def system_role
 class ShallowUserSerializer < ActiveModel::Serializer
   attributes :id, :name, :email
 end
+
+class HelpdeskUserSerializer < ActiveModel::Serializer
+  attributes :id, :name, :email, :units_taught
+
+  def units_taught
+    ActiveModel::ArraySerializer.new(object.units_taught, each_serializer: ShallowUnitSerializer)
+  end
+end
```

15 db/migrate/20160705025331_create_helpdesk_tickets.rb View

```
@@ -0,0 +1,15 @@
+class CreateHelpdeskTickets < ActiveRecord::Migration
+  def change
+    create_table :helpdesk_tickets do |t|
+      t.references :project, null: false
+      t.references :task, null: true
+
+      t.string :comments, null: true
+      t.boolean :is_resolved, null: false, default: false
+
+      t.timestamps
+    end
+    add_index :helpdesk_tickets, :project_id
+    add_index :helpdesk_tickets, :task_id
+  end
+end
```

6 db/migrate/20160708011422_change_helpdesk_ticket_comment_to_description.rb View

```
@@ -0,0 +1,6 @@
+class ChangeHelpdeskTicketCommentToDescription < ActiveRecord::Migration
+  def change
+    rename_column :helpdesk_tickets, :comments, :description
+    change_column :helpdesk_tickets, :description, :string, :limit => 2048
+  end
+end
```

11 db/migrate/20160722114921_create_helpdesk_sessions.rb View

```
@@ -0,0 +1,11 @@
+class CreateHelpdeskSessions < ActiveRecord::Migration
+  def change
+    create_table :helpdesk_sessions do |t|
+      t.references :user, null: false
+      t.datetime :clock_on_time, null: false
+      t.datetime :clock_off_time, null: false
+      t.timestamps
+    end
+    add_index :helpdesk_sessions, :user_id
```

```
+  end
+end
```

5 db/migrate/20160725141011_add_resolved_at_to_helpdesk_ticket.rb View

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

5 db/migrate/20160725144833_add_minutes_to_resolve_to_helpdesk_ticket.rb View

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

5 db/migrate/20160826233551_add_is_closed_to_helpdesk_ticket.rb View

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

5 db/migrate/20160910065932_rename_resolved_at_to_closed_at.rb View

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

27 db/schema.rb View

```
@@ -11,7 +11,7 @@
 #
 # It's strongly recommended that you check this file into your version control system.

-ActiveRecord::Schema.define(version: 20160503222451) do
+ActiveRecord::Schema.define(version: 20160910065932) do

   # These are extensions that must be enabled in order to support this database
   enable_extension "plpgsql"
@@ -74,6 +74,31 @@

   add_index "helpdesk_schedules", ["user_id"], name: "index_helpdesk_schedules_on_user_id", using: :btree

+  create_table "helpdesk_sessions", force: :cascade do |t|
+    t.integer  "user_id",        null: false
+    t.datetime "clock_on_time",  null: false
+    t.datetime "clock_off_time", null: false
+    t.datetime "created_at"
+    t.datetime "updated_at"
+  end
+
+  add_index "helpdesk_sessions", ["user_id"], name: "index_helpdesk_sessions_on_user_id", using: :btree
```

```
+
+  create_table "helpdesk_tickets", force: :cascade do |t|
+    t.integer  "project_id",                                  null: false
+    t.integer  "task_id"
+    t.string   "description",        limit: 2048
+    t.boolean  "is_resolved",                    default: false, null: false
+    t.datetime "created_at"
+    t.datetime "updated_at"
+    t.datetime "closed_at"
+    t.float    "minutes_to_resolve"
+    t.boolean  "is_closed",                      default: false
+  end
+
+  add_index "helpdesk_tickets", ["project_id"], name: "index_helpdesk_tickets_on_project_id", using: :btree
+  add_index "helpdesk_tickets", ["task_id"], name: "index_helpdesk_tickets_on_task_id", using: :btree
+
   create_table "learning_outcome_task_links", force: :cascade do |t|
     t.text     "description"
     t.integer  "rating"
```

131 lib/helpers/database_populator.rb View

```
@@ -2,6 +2,7 @@
 require 'faker'
 require 'bcrypt'
 require 'json'
+require_all 'lib/helpers'

 #
 # This class populates data in the database
@@ -29,7 +30,7 @@ def initialize(scale = :small)
         some_tutorials: 1,
         many_tutorials: 1,
         max_tutorials: 4,
-        tickets_generated: 10
+        tickets_to_generate: 30
       },
       large: {
         min_students: 15,
@@ -41,7 +42,7 @@ def initialize(scale = :small)
         some_tutorials: 2,
         many_tutorials: 4,
         max_tutorials: 20,
-        tickets_generated: 50
+        tickets_to_generate: 100
       }
     }
     accepted_scale_types = scale_data.keys
@@ -135,14 +136,80 @@ def generate_units
       generate_and_align_ilos_for_unit(unit, unit_details)
       generate_tutorials_and_enrol_students_for_unit(unit, unit_details)
     end
+    puts "!"
   end

   #
-  # Random project helper
+  # Generates some units
   #
-  def random_project
-    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 "!"
213    end
214
215    private
@@ -323,42 +390,22 @@ def generate_tutorials_and_enrol_students_for_unit(unit, unit_details)
390    end
391
392    #
-  # Finds or creates a student if not found
-  #
-  def find_or_create_student(username)
-    unless @user_cache.has_key?(username)
-      profile = {
-        first_name:             Faker::Name.first_name,
-        last_name:              Faker::Name.last_name,
-        nickname:               username,
-        role_id:                Role.student_id,
-        email:                  "#{username}@doubtfire.com",
-        username:               username,
-        password:               'password',
-        password_confirmation:  'password'
-      }
-      user = User.create!(profile)
-      @user_cache[username] = user
-    end
-    @user_cache[username]
-  end
-
-  #
393    # Generated fixed data here for students and units
394    #
395    def generate_fixed_data
396      # Define fixed user data here
397      @user_data = {
398        acain:              {first_name: "Andrew",  last_name: "Cain",          nickname: "Macite",         role_id: Role
-       cwoodward:          {first_name: "Clinton", last_name: "Woodward",      nickname: "The Giant",      role_id: Role
-       ajones:             {first_name: "Allan",   last_name: "Jones",         nickname: "P-Jiddy",        role_id: Role
399 +       aconvenor:          {first_name: "Clinton", last_name: "Woodward",      nickname: "The Giant",      role_id: Role
400 +       aadmin:             {first_name: "Allan",   last_name: "Jones",         nickname: "P-Jiddy",        role_id: Role
401        rwilson:            {first_name: "Reuben",  last_name: "Wilson",        nickname: "Reubs",          role_id: Role
-       akihironoguchi:     {first_name: "Akihiro", last_name: "Noguchi",       nickname: "Animations",     role_id: Role
402 +       atutor:             {first_name: "Akihiro", last_name: "Noguchi",       nickname: "Animations",     role_id: Role
403        acummaudo:          {first_name: "Alex",    last_name: "Cummaudo",      nickname: "DoubtfireDude",  role_id: Role
404        cliff:              {first_name: "Cliff",   last_name: "Warren",        nickname: "Cliff",          role_id: Role
405        joostfunkekupper:   {first_name: "Joost",   last_name: "Funke Kupper",  nickname: "Joe",            role_id: Role
406        angusmorton:        {first_name: "Angus",   last_name: "Morton",        nickname: "Angus",          role_id: Role
-       "123456X" =>        {first_name: "Fred",    last_name: "Jones",         nickname: "Foo",            role_id: Role
407 +       "123456X" =>        {first_name: "Fred",    last_name: "Jones",         nickname: "Foo",            role_id: Role
408 +       astudent:           {first_name: "student", last_name: "surname",       nickname: "Foo",            role_id: Role
409      }
410      # Add 10 tutors to fixed info
411      10.times do |count|
@@ -381,14 +428,14 @@ def generate_fixed_data
428        intro_prog: {
429          code: "COS10001",
430          name: "Introduction to Programming",
-        convenors: [ :acain, :cwoodward ],
431 +        convenors: [ :acain, :aconvenor ],
```

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

22 lib/helpers/find_or_create_students.rb View

```
...  ...  @@ -0,0 +1,22 @@
       1  +#
```

```
+# Finds or creates a student if not found
+#
+def find_or_create_student(username)
+  user_created = nil
+  using_cache = !!@user_cache
+  if not using_cache or not @user_cache.has_key?(username)
+    profile = {
+      first_name:            Faker::Name.first_name,
+      last_name:             Faker::Name.last_name,
+      nickname:              username,
+      role_id:               Role.student_id,
+      email:                 "#{username}@doubtfire.com",
+      username:              username,
+      password:              'password',
+      password_confirmation:  'password'
+    }
+    user_created = User.create!(profile)
+    @user_cache[username] = user_created if using_cache
+  end
+  user_created || @user_cache[username]
+end
```

25 lib/helpers/randomizer.rb View

```
@@ -0,0 +1,25 @@
+class Randomizer
+  #
+  # Randomly returns a new model (e.g., random_record_for_model(Project))
+  #
+  def self.random_record_for_model(model)
+    throw "Must provide a model" unless model.is_a? Class
+    id = model.all.pluck(:id).sample
+    model.find(id)
+  end
+
+  #
+  # Randomly returns a new task for the given project
+  #
+  def self.random_task_for_project(project)
+    task_def = self.random_task_def_for_project(project)
+    project.task_for_task_definition(task_def)
+  end
+
+  #
+  # Randomly returns a new task definition for the given project
+  #
+  def self.random_task_def_for_project(project)
+    project.unit.task_definitions.sample
+  end
+end
```

26 lib/tasks/populate.rake View

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

```
-       profile = {
-         first_name:            Faker::Name.first_name,
-         last_name:             Faker::Name.last_name,
-         nickname:              username,
-         role_id:               Role.student_id,
-         email:                 "#{username}@doubtfire.com",
-         username:              username,
-         password:              'password',
-         password_confirmation: 'password'
-       }
-
-       result = User.create!(profile)
-       return result
-     end
-   }
-
    unit = Unit.first
    tutes = unit.tutorials
    for student_count in 0..2000
-     proj = unit.enrol_student(find_or_create_student.call("student_#{student_count}"), tutes[student_count % tutes.co
+     student = find_or_create_student("student_#{student_count}")
+     proj = unit.enrol_student(student, tutes[student_count % tutes.count])
    end
  end

@@ -177,6 +156,7 @@ namespace :db do
    dbpop = DatabasePopulator.new ENV['SCALE']
    dbpop.generate_users()
    dbpop.generate_units()
+   dbpop.generate_helpdesk_tickets()

    # Run simulate signoff?
    unless !args.nil? && args[:extend_populate]
```

2 lib/tasks/test_setup.rake View

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

17 test/api/auth_test.rb View

```
@@ -1,6 +1,6 @@
 require 'test_helper'

-class AuthTest < MiniTest::Test
+class AuthTest < ActiveSupport::TestCase
   include Rack::Test::Methods
   include TestHelpers::AuthHelper
   include TestHelpers::JsonHelper
@@ -38,10 +38,7 @@ def test_auth_post
    # These match the model object... so can compare in loops
    user_keys = [ 'id', 'email', 'first_name', 'last_name', 'username', 'nickname', 'receive_task_notifications', 'rece
```

```
-    assert_json_matches_model(response_user_data, expected_auth, user_keys)
-
-    user_keys.each { |k| assert response_user_data.has_key?(k), "Response has key #{k}" }
-    user_keys.each { |k| assert_equal expected_auth[k], response_user_data[k], "Values for key #{k} match" }
+    assert_json_matches_model(response_user_data, expected_auth, *user_keys)

     # Check other values returned
     assert_equal expected_auth.name, response_user_data['name'], 'Names match'
@@ -72,28 +69,28 @@ def test_auth_roles
       {
         expect: Role.admin,
         post: {
-            username: "acain",
+            username: "aadmin",
             password: "password"
         }
       },
       {
         expect: Role.convenor,
         post: {
-            username: "jrenzella",
+            username: "aconvenor",
             password: "password"
         }
       },
       {
         expect: Role.tutor,
         post: {
-            username: "rwilson",
+            username: "atutor",
             password: "password"
         }
       },
       {
         expect: Role.student,
         post: {
-            username: "acummaudo",
+            username: "astudent",
             password: "password"
         }
       }
@@ -116,7 +113,6 @@ def test_auth_roles

   # Test put for authentication token
   def test_auth_put
-    auth_token = get_auth_token
     data_to_put = {
         username: "acain",
         password: "password"
@@ -136,7 +132,6 @@ def test_auth_put

   # Test for deleting authentication token
   def test_auth_delete
-    auth_token = get_auth_token
     # Get the auth token needed for delete test
     delete "/api/auth/#{auth_token}.json", "CONTENT_TYPE" => 'application/json'
     # 200 response code means success!
```

95 test/api/helpdesk/ticket_test.rb View

```
... ... @@ -0,0 +1,95 @@
```

```
+require 'test_helper'
+
+class TicketsTest < ActiveSupport::TestCase
+  include Rack::Test::Methods
+  include TestHelpers::AuthHelper
+  include TestHelpers::JsonHelper
+
+  def app
+    Rails.application
+  end
+
+  # POST /helpdesk/tickets
+  def test_post_helpdesk_ticket
+    project = Randomizer.random_record_for_model(Project)
+
+    # Can't create ticket if this user already has a ticket
+    while HelpdeskTicket.user_has_ticket_open? project.user_id
+      project = Randomizer.random_record_for_model(Project)
+    end
+
+    ticket = {
+      project_id: project.id,
+      description: Populator.words(5..10),
+      task_definition_id: rand > 0.33 ? Randomizer.random_task_def_for_project(project).id : nil
+    }
+
+    data_to_post = add_auth_token ticket, project.user
+
+    post_json '/api/helpdesk/tickets', data_to_post
+
+    # Last ticket ids should match
+    expected_ticket = HelpdeskTicket.last
+    assert_json_matches_model last_response_body, expected_ticket, :id
+  end
+
+  # GET /helpdesk/tickets
+  def test_get_all_helpdesk_tickets
+    get with_auth_token '/api/helpdesk/tickets'
+    expect = ActiveModel::ArraySerializer.new(HelpdeskTicket.all, each_serializer: ShallowHelpdeskTicketSerializer)
+    assert_json_equal expect, last_response_body
+  end
+
+  # GET /helpdesk/tickets
+  def test_get_all_unresolved_helpdesk_tickets
+    get with_auth_token '/api/helpdesk/tickets?filter=unresolved'
+    expect = ActiveModel::ArraySerializer.new(HelpdeskTicket.all_unresolved, each_serializer: ShallowHelpdeskTicketSeri
+    assert_json_equal expect, last_response_body
+  end
+
+  # GET /helpdesk/tickets?filter=resolved
+  def test_get_resolved_helpdesk_tickets
+    get with_auth_token '/api/helpdesk/tickets?filter=resolved'
+    expect = ActiveModel::ArraySerializer.new(HelpdeskTicket.all_resolved, each_serializer: ShallowHelpdeskTicketSerial
+    assert_json_equal expect, last_response_body
+  end
+
+  # GET /helpdesk/tickets?user_id=1&filter=resolved
+  def test_get_resolved_helpdesk_tickets_for_user
+    id = 1
+    get with_auth_token "/api/helpdesk/tickets?user_id=#{id}&filter=resolved"
+    expect = ActiveModel::ArraySerializer.new(HelpdeskTicket.all_resolved(id), each_serializer: ShallowHelpdeskTicketSe
+    assert_json_equal expect, last_response_body
+  end
```

```
+
+  # GET /helpdesk/tickets/:id
+  # id = 1
+  def test_get_helpdesk_ticket_with_id
+    id = 1
+    get with_auth_token "/api/helpdesk/tickets/#{id}"
+    expect = HelpdeskTicketSerializer.new HelpdeskTicket.find(id)
+    assert_json_equal expect, last_response_body
+  end
+
+  # DELETE /helpdesk/tickets/:id?resolve=false
+  def test_delete_a_ticket_by_closing
+    ticket = HelpdeskTicket.all_unresolved.first
+    delete with_auth_token "/api/helpdesk/tickets/#{ticket.id}"
+    # Reload the ticket from DB with updated info
+    ticket = HelpdeskTicket.find(ticket.id)
+    assert ticket.is_closed
+    refute ticket.is_resolved
+  end
+
+  # DELETE /helpdesk/tickets/:id?resolve=true
+  def test_delete_a_ticket_by_resolving
+    ticket = HelpdeskTicket.all_unresolved.first
+    delete with_auth_token "/api/helpdesk/tickets/#{ticket.id}?resolve=true"
+    # Reload the ticket from DB with updated info
+    ticket = HelpdeskTicket.find(ticket.id)
+    assert ticket.is_resolved
+    assert ticket.is_closed
+    dt = DateTime.parse(last_response_body['closed_at']).inspect
+    assert dt, ticket.closed_at
+  end
+end
```

12 test/api/units_test.rb View

```
@@ -1,7 +1,7 @@
 require 'test_helper'
 require 'date'

-class UnitsTest < MiniTest::Test
+class UnitsTest < ActiveSupport::TestCase
   include Rack::Test::Methods
   include TestHelpers::AuthHelper
   include TestHelpers::JsonHelper
@@ -10,10 +10,6 @@ def app
     Rails.application
   end

-  def setup
-    @auth_token = get_auth_token()
-  end
-
   # ------------------------------------------------------------------------- #
   # --- Endpoint testing for:
   # ------- /api/units.json
@@ -100,7 +96,7 @@ def test_units_post_tutorial
   # Test GET for getting all units
   def test_units_get
     # The GET we are testing
-    get "/api/units.json?auth_token=#{@auth_token}"
+    get with_auth_token "/api/units.json"

     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}"
        123 +   get with_auth_token "/api/units/1.json"
128 124
129 125     actual_unit = JSON.parse(last_response.body)
130 126     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}"
        136 +   get with_auth_token "/api/units/2.json"
141 137
142 138     actual_unit = JSON.parse(last_response.body)
143 139     expected_unit = Unit.find(2)
```

33 test/helpers/auth_helper.rb View

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

```
+    # Alias for above for nicer usage (e.g., get with_auth_token "http://")
+    #
+    def with_auth_token(data, user = User.first)
+      add_auth_token data, user
+    end
+
+    module_function :auth_token
     module_function :add_auth_token
+    module_function :with_auth_token
   end
 end
```

33 test/helpers/json_helper.rb View

```
@@ -20,15 +20,40 @@ def put_json(endpoint, data)
     end

     #
-    # Assert that a JSON response matches the model and keys provided
+    # Assert that a JSON response matches the model and keys provided.
+    # Keys MUST be symbols
     #
-    def assert_json_matches_model(response_json, model, keys)
-      keys.each { |k| assert response_json.has_key?(k), "Response has key #{k}"}
-      keys.each { |k| assert_equal model[k], response_json[k], "Values for key #{k} match" }
+    def assert_json_matches_model(response_json, model, *keys)
+      keys.each { |k| assert response_json.key?(k.to_s), "Response has key #{k}"}
+      keys.each { |k| assert_equal model[k], response_json[k.to_s], "Values for key #{k} match" }
     end

+    #
+    # Last response body parsed from JSON
+    #
+    def last_response_body
+      JSON.parse(last_response.body)
+    end
+
+    #
+    # Converts from an ActiveRelation to JSON (without Ruby objects inside the hash)
+    #
+    def json_hashed(hash)
+      JSON.parse(hash.to_json)
+    end
+
+    #
+    # Assert that the lefthand matches the right-hand as json hash
+    #
+    def assert_json_equal(lhs, rhs)
+      assert_equal json_hashed(lhs), json_hashed(rhs)
     end

     module_function :assert_json_matches_model
     module_function :post_json
     module_function :put_json
+    module_function :last_response_body
+    module_function :json_hashed
+    module_function :assert_json_equal
   end
 end
```

68 test/models/helpdesk/ticket_test.rb View

```
@@ -0,0 +1,68 @@
```

```
+require "test_helper"
+
+class HelpdeskTicketTest < ActiveSupport::TestCase
+  setup do
+    @test_desc = 'Hello, World!'
+    @test_project = Randomizer.random_record_for_model(Project)
+    @test_task = Randomizer.random_task_for_project(@test_project)
+    @tickets = {
+      without_task: HelpdeskTicket.create!(
+        project: @test_project
+      ),
+      with_task: HelpdeskTicket.create!(
+        project: @test_project,
+        task: @test_task
+      ),
+      with_description: HelpdeskTicket.create!(
+        project: @test_project,
+        description: @test_desc
+      )
+    }
+  end
+
+  test "helpdesk tickets have projects" do
+    @tickets.values.each do |t|
+      assert t.project, @test_project
+    end
+  end
+
+  test "only one helpdesk ticket has a description" do
+    assert_not_nil @tickets[:with_description].description
+    assert_equal @tickets[:with_description].description, @test_desc
+    assert_nil @tickets[:with_task].description
+    assert_nil @tickets[:without_task].description
+  end
+
+  test "only one helpdesk ticket has a task" do
+    assert_equal @tickets[:with_task].task, @test_task
+    assert_nil @tickets[:with_description].task
+    assert_nil @tickets[:without_task].task
+  end
+
+  test "resolving a helpdesk ticket" do
+    ticket_to_resolve = @tickets[:without_task]
+    ticket_to_resolve.resolve
+    assert ticket_to_resolve, true
+    # both without_task and ticket_to_resolve should match
+    assert_equal @tickets[:without_task], ticket_to_resolve
+    # others should remain false
+    refute @tickets[:with_task].is_resolved
+    refute @tickets[:with_description].is_resolved
+  end
+
+  test "the task? method should be true if the task has a task" do
+    assert @tickets[:with_task].task?
+  end
+
+  test "the student method should match the associated project's student" do
+    assert_equal @tickets[:with_task].student, @test_project.student
+    assert_equal @tickets[:with_task].student, @test_project.student
+    assert_equal @tickets[:with_description].student, @test_project.student
+  end
+
+  test "the unit method should match the associated project's unit" do
```

```
+    assert_equal @tickets[:with_task].unit, @test_project.unit
+    assert_equal @tickets[:with_task].unit, @test_project.unit
+    assert_equal @tickets[:with_description].unit, @test_project.unit
+  end
+end
```

18 test/test_helper.rb View

```
@@ -17,16 +17,14 @@
 # Require minitest extensions
 require 'minitest/rails'
 require 'minitest/pride'
-require 'minitest/autorun'
 require 'minitest/osx'
+require 'minitest/around'

-# Require all test helpers
-require_all 'test/helpers'
+# Require test helpers
 require 'rails/test_help'
 require 'database_cleaner'

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

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

-  # After setup of all test, start database cleaner to undo transactions
-  def before_teardown
-    super
+  def setup
     DatabaseCleaner.start
   end

-  # After teardown
-  def after_teardown
+  def teardown
     DatabaseCleaner.clean
-    super
   end

   # Add more helper methods to be used by all tests here...
+  require_all 'test/helpers'
+
+  extend MiniTest::Spec::DSL
+
 end
```

4 Doubtfire Web Source Code Changes

See attached document

2 Gruntfile.js View

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

```
+ #
+ HelpdeskSession.prototype.clockOff = (callback) ->
+   HelpdeskSession.clockOffSession(this.id, callback)
+
+ #
+ # Returns the duration until the session expires
+ #
+ HelpdeskSession.prototype.timeUntilFinish = ->
+   clockOffTime = moment(this.clock_off_time)
+   moment.duration(clockOffTime.diff())
+
+ #
+ # Returns the currently working session for the user id provided, or
+ # null if no such session exists
+ #
+ HelpdeskSession.currentWorkingSession = (userId, callback) ->
+   HelpdeskSession.getActiveSessions userId, (error, success) ->
+     return callback(error) if error
+     callback(null, success[0])
+
+ HelpdeskSession
+)
```

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

```
@@ -0,0 +1,21 @@
+angular.module("doubtfire.api.models.helpdesk-stats", [])
+#
+# API endpoint for getting helpdesk stats
+#
+.factory("HelpdeskStats", ($http, $rootScope, $interval, api) ->
+  HelpdeskStats = {}
+
+  #
+  # Gets the helpdesk statistics from the time specified by `from` to the
+  # time specified by `to`.
+  #
+  # If to is null, statistics will fetch to now.
+  # If from is null, all statistics from the start of time are returned.
+  #
+  HelpdeskStats.get = (from, to, interval, callback) ->
+    onSuccess = (response) -> callback(null, response.data)
+    onFailure = (response) -> callback(response)
+    $http.get("#{api}/helpdesk/stats/dashgraph").then(onSuccess, onFailure)
+
+  HelpdeskStats
+)
```

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

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

```
+  #
+  HelpdeskTicket.submitTicket = (projectId, description, taskDefinitionId, callback) ->
+    dataToPost =
+      project_id: projectId
+      description: description
+      task_definition_id: taskDefinitionId
+    onSuccess = (response) -> callback(null, response)
+    onFailure = (response) -> callback(response)
+    HelpdeskTicket.create(dataToPost)
+                  .$promise
+                  .then(onSuccess, onFailure)
+
+  #
+  # Gets all tickets by a specific state, optionally for the specified user id.
+  # If user is not specified, then all tickets will be returned.
+  #
+  HelpdeskTicket.getTicketsByState = (state, userId, callback) ->
+    onSuccess = (response) -> callback(null, response)
+    onFailure = (response) -> callback(response)
+    HelpdeskTicket.query({ user_id: userId, shallow: userId?, filter: state })
+                  .$promise
+                  .then(onSuccess, onFailure)
+
+  #
+  # Gets all unresolved tickets (optionally by user)
+  #
+  HelpdeskTicket.getUnresolvedTickets = (userId, callback) ->
+    HelpdeskTicket.getTicketsByState('unresolved', userId, callback)
+
+  #
+  # Gets all resolved tickets (optionally by user)
+  #
+  HelpdeskTicket.getResolvedTickets = (userId, callback) ->
+    HelpdeskTicket.getTicketsByState('resolved', userId, callback)
+
+  #
+  # Gets all closed tickets (optionally by user)
+  #
+  HelpdeskTicket.getClosedTickets = (userId, callback) ->
+    HelpdeskTicket.getTicketsByState('closed', userId, callback)
+
+  #
+  # Closes a specified ticket with the given ID
+  #
+  HelpdeskTicket.closeTicket = (ticketId, callback) ->
+    onSuccess = (response) -> callback(null, response)
+    onFailure = (response) -> callback(response)
+    HelpdeskTicket.delete({ id: ticketId })
+                  .$promise
+                  .then(onSuccess, onFailure)
+
+  #
+  # Prototype close ticket method
+  #
+  HelpdeskTicket.prototype.close = (callback) ->
+    HelpdeskTicket.closeTicket(this.id, callback)
+
+  #
+  # Resolves a specified ticket with the given ID
+  #
+  HelpdeskTicket.resolveTicket = (ticketId, callback) ->
+    onSuccess = (response) -> callback(null, response)
+    onFailure = (response) -> callback(response)
```

```
+    HelpdeskTicket.delete({ id: ticketId, resolve: true })
+                  .$promise
+                  .then(onSuccess, onFailure)
+
+  #
+  # Prototype resolve ticket method
+  #
+  HelpdeskTicket.prototype.resolve = (callback) ->
+    HelpdeskTicket.resolveTicket(this.id, callback)
+
+  #
+  # Returns the duration since the ticket was open
+  #
+  HelpdeskTicket.prototype.lengthOfTimeOpen = ->
+    createdTime = moment(this.created_at)
+    moment.duration(createdTime.diff())
+
+  #
+  # Checks if the user with the specified ticket has a ticket open already
+  # and if so returns that ticket in the callback
+  #
+  HelpdeskTicket.currentOpenTicket = (userId, callback) ->
+    HelpdeskTicket.getUnresolvedTickets userId, (error, success) ->
+      return callback(error) if error
+      callback(null, success[0])
+
+  HelpdeskTicket
+)
```

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

```
@@ -8,6 +8,9 @@ angular.module("doubtfire.api.models", [
   "doubtfire.api.models.tutorial"
   "doubtfire.api.models.learning-alignments"
   "doubtfire.api.models.intended-learning-outcome"
+  "doubtfire.api.models.helpdesk-ticket"
+  "doubtfire.api.models.helpdesk-session"
+  "doubtfire.api.models.helpdesk-stats"
   "doubtfire.api.models.task"
   "doubtfire.api.models.task-comment"
   "doubtfire.api.models.task-definition"
```

1 src/app/app.coffee View

```
@@ -16,6 +16,7 @@ angular.module('doubtfire', [
   'doubtfire.tasks'
   'doubtfire.projects'
   'doubtfire.users'
+  'doubtfire.helpdesk'
   'doubtfire.groups'
   'doubtfire.visualisations'
 ])
```

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

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

```
+])
 .controller("BasicHeaderCtrl", ($scope, $state, $modal, User, AboutDoubtfireModal, UserNotificationSettingsModal, UserS
   $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 {
   }
 }
+
+@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>
   <!-- Collect the nav links, forms, and other content for toggling -->
   <div class="collapse navbar-collapse" ng-class="'!navCollapsed && 'in'" ng-show="'!navCollapsed">
```

```
17  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'}">
23          -           <a href="{{link.url}}">{{link.name}}</a>
24          -         </li>
25          -       </ul>
26          -     </li>
            18  +     <helpdesk-header
            19  +       projects="projects"
            20  +       unit-roles="unitRoles">
            21  +     </helpdesk-header>
27  22
28  23      <li class="dropdown dropdown-units" dropdown ng-hide="projects.length == 0 && unitRoles.length == 0">
29  24        <a role="button" class="dropdown-toggle" dropdown-toggle><span class="glyphicon glyphicon-book"></span> Units <
```

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, HelpdeskSess
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 dropown 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
```

```
+    $rootScope.$on 'CurrentOpenTicket', (event, ticket) ->
+      $scope.currentOpenTicket = ticket
+    $rootScope.$on 'CurrentWorkingSession', (event, session) ->
+      $scope.currentWorkingSession = session
+
+    #
+    # Opens the submit ticket modal
+    #
+    $scope.openHelpdeskTicketModal = ->
+      HelpdeskTicketModal.show $scope.currentOpenTicket
+
+    #
+    # Opens the session modal
+    #
+    $scope.openHelpdeskSessionModal = ->
+      HelpdeskSessionModal.show $scope.currentWorkingSession
```

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

```
@@ -0,0 +1,20 @@
+// Mobile responsive fixes
+@media (min-width: $screen-md) {
+  .helpdesk.header {
+    display: inline;
+    .helpdesk-direct-link {
+      padding-right: 0;
+      .badge {
+        background-color: $doubtfire-color;
+      }
+    }
+    .dropdown-toggle {
+      float: right;
+    }
+  }
+}
+@media (max-width: $screen-md) {
+  .helpdesk.header .helpdesk-direct-link {
+    display: none;
+  }
+}
```

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

```
@@ -0,0 +1,33 @@
+<li dropdown class="helpdesk header" ng-show="showHelpdeskDropdown">
+  <a class="helpdesk-direct-link pull-left" ng-click="openHelpdeskTicketModal()" ng-if="currentOpenTicket">
+    <span class="badge" tooltip="Ticket open for {{currentOpenTicket.lengthOfTimeOpen().humanize()}}" tooltip-placement
+      <i class="fa fa-ticket"></i>
+    </span>
+  </a>
+  <a class="helpdesk-direct-link pull-left" ng-click="openHelpdeskSessionModal()" ng-if="currentWorkingSession">
+    <span class="badge" tooltip="You finish working in {{currentWorkingSession.timeUntilFinish().humanize()}}" tooltip-
+      <i class="fa fa-clock-o"></i>
+    </span>
+  </a>
+  <a class="dropdown-toggle" dropdown-toggle>
+    <i class="fa fa-ambulance"></i>
+    Helpdesk
+    <b class="caret"></b>
+  </a>
+  <ul class="dropdown-menu">
+    <li ng-if="studiesUnit">
+      <a ng-click="openHelpdeskTicketModal()">
```

```
+        {{ currentOpenTicket ? "View Open Ticket" : "Submit a Ticket" }}
+      </a>
+    </li>
+    <li ng-if="teachesUnit">
+      <a ng-click="openHelpdeskSessionModal()">
+        {{ currentWorkingSession ? "Clock Off" : "Clock On" }}
+      </a>
+    </li>
+    <li class="divider"></li>
+    <li>
+      <a ui-sref="helpdesk"> View Helpdesk Dashboard </a>
+    </li>
+  </ul>
+</li>
```

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

```
@@ -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

```
@@ -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
```

```
+  #
+  projectService.findProject = (id, callback) ->
+    projectService.getProjects (projects) ->
+      projFound = _.find(projects, (p) -> p.project_id is id)
+      callback(projFound)
+
   ###
   projects's can update their task stats
   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", [])
           (response) ->
             alertService.add("danger", "Grade was not updated: #{response.data.error}", 8000)

-      projectService.updateTaskStats(student, student.stats)
+      projectService.updateTaskStats(student, student.stats) if student.stats
       projectService.addTaskDetailsToProject(student, unit)

     unit.getGroups = (group_set, group_callback) ->
```

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

```
@@ -0,0 +1,33 @@
+angular.module('doubtfire.helpdesk.helpdesk-ticket', [])
+#
+# Directive to display information related to a helpdesk ticket
+#
+.directive('helpdeskTicket', ->
+  restrict: 'E'
+  templateUrl: 'helpdesk/helpdesk-ticket/helpdesk-ticket.tpl.html'
+  replace: true
+  scope:
+    ticket: '='
+  controller: ($scope, HelpdeskTicket, projectService, unitService, HelpdeskTicketModal) ->
+
+    #
+    # Returns warning style for ticket
+    #
+    $scope.warningStyle = ->
+      minsWaiting = -$scope.ticket.lengthOfTimeOpen().asMinutes()
+      if minsWaiting < 3
+        'success'
+      else if 3 <= minsWaiting < 6
+        'primary'
+      else if 6 <= minsWaiting < 9
+        'warning'
+      else if minsWaiting >= 9
+        'danger'
+
+    #
+    # Opens the submit ticket modal
+    #
+    $scope.openHelpdeskTicketModal = ->
+      return if $scope.isLoading
+      HelpdeskTicketModal.show $scope.ticket
+)
```

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

```
@@ -0,0 +1,49 @@
+.helpdesk-ticket {
+  $br: 8px;
+  font-size: 1.3em;
+  // Mobile responsive fixes
+  @media (max-width: $screen-md) {
+    margin-bottom: 1em;
+    .unit, .unit-name, .task-name {
+      text-align: center !important;
+      width: 100% !important;
+    }
+  }
+  header, .content {
+    padding: 8px;
+  }
+  header {
+    background-color: $doubtfire-color;
+    color: #fff;
+    font-size: 1.25em;
+    border-radius: $br $br 0 0;
+    line-height: 1.25;
+  }
+  .content {
+    color: #525252;
+    background-color: #ecedef;
+    .unit-name {
+      text-overflow: ellipsis;
+      width: 65%;
+      text-align: right;
+      overflow: hidden;
+      white-space: nowrap;
+    }
+  }
+  .content .loading {
+    text-align: center;
+    font-size: 2em;
+  }
+  hr {
+    border: 1px dashed grey;
+  }
+  border: 2px solid #ecedef;
+  border-radius: $br;
+  transition: all 0.25s ease;
+  &:hover {
+    position: relative;
+    top: -10px;
+    cursor: pointer;
+    box-shadow: 0px 0px 10px 0px rgba(0,0,0,0.4);
+  }
+}
```

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

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

```
+      <i class="fa fa-spinner fa-pulse"></i>
+    </div><!--/loading-ticket-->
+    <div ng-if="!isLoading">
+      <div class="unit">
+        <label class="label label-info">{{ticket.project.unit_code}}</label>
+        <strong class="unit-name pull-right">{{ticket.project.unit_name}}</strong>
+      </div>
+      <div class="task-name text-right">
+        {{ticket.task_definition.name}}
+      </div>
+      <hr>
+      <div class="text-center">
+        Waiting for
+        <strong class="text-{{warningStyle()}}">
+          {{ticket.lengthOfTimeOpen().humanize()}}
+        </strong>
+      </div>
+    </div><!--/data-loaded-->
+  </div><!--/ticket-content-->
+</div>
```

5 src/app/helpdesk/helpdesk.coffee View

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

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

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

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

```
@@ -0,0 +1,81 @@
+#
+# The helpdesk session modal is the modal by which staff can clock on and clock
+# off from the helpdesk
+#
+angular.module('doubtfire.helpdesk.modals.session-modal', [])
+
+.factory('HelpdeskSessionModal', ($modal) ->
+  HelpdeskSessionModal = {}
+
+  #
+  # Pass in the user, project
+  #
+  HelpdeskSessionModal.show = (session) ->
+    $modal.open
+      templateUrl: 'helpdesk/modals/session-modal/session-modal.tpl.html'
+      controller: 'HelpdeskSessionModal'
+      resolve:
+        session: -> session
+
+  HelpdeskSessionModal
+)
+
```

```
+.controller('HelpdeskSessionModal', ($scope, $state, $rootScope, $interval, $modalInstance, HelpdeskSession, Confirmati
+  $scope.clockOffState = session? # I can clock off if I was given a session
+  $scope.clockOnState = !session? # I can clock on if I wasn't given a session
+
+  # Strip the profile
+  $scope.currentUser = currentUser.profile
+
+  # Duration to clock on
+  $scope.duration = { hours: null }
+
+  #
+  # Updates the automatic clock off time
+  #
+  updateAutomaticClockOffTime = (newDuration) ->
+    return unless newDuration?
+    $scope.automaticClockOffTime =
+      moment().add($scope.duration.hours, 'hours')
+
+  # Watch duration to calculate the automaticClockOffTime
+  $scope.$watch 'duration.hours', updateAutomaticClockOffTime
+
+  # If we have a session, work out the automatic clock off time
+  if $scope.clockOffState
+    $scope.automaticClockOffTime = moment(session.clock_off_time)
+    updateTimeLeft = ->
+      $scope.timeLeft = moment.duration($scope.automaticClockOffTime.diff())
+    updateTimeLeft()
+    updateTimeLeftInterval = $interval updateTimeLeft, 1000
+    # Must explicitly destroy the time interval when the controller ends!
+    $scope.$on '$destroy', ->
+      $interval.cancel(updateTimeLeftInterval)
+
+  #
+  # Clocks on the session
+  #
+  $scope.clockOn = ->
+    clockOnCallback = (error, success) ->
+      if success
+        $modalInstance.close(success)
+        $rootScope.$broadcast('CurrentWorkingSession', success)
+        alertService.add("success", "Clocked on successfully.", 2000)
+      if error
+        alertService.add("danger", "Error: #{error.data.error}", 6000)
+    HelpdeskSession.clockOn(currentUser.id, $scope.duration.hours, clockOnCallback)
+
+  #
+  # Clocks off the session
+  #
+  $scope.clockOff = ->
+    clockOffCallback = (error, success) ->
+      if success
+        $modalInstance.close(success)
+        $rootScope.$broadcast('CurrentWorkingSession', null)
+        alertService.add("success", "You have been clocked off.", 2000)
+      if error
+        alertService.add("danger", "Error: #{error.data.error}", 6000)
+    ConfirmationModal.show 'Clock off now', "You still have #{session.timeUntilFinish().humanize()} left to work. Are y
+      session.clockOff(clockOffCallback)
+)
```

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

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

```
+.session-modal {
+  .duration-entry {
+    margin: 1em auto;
+  }
+  .time-left {
+    font-size: 3em;
+  }
+}
```

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

```
@@ -0,0 +1,62 @@
+<div class="session-modal">
+  <form class="form-horizontal" name="sessionForm" role="form">
+    <div class="modal-header">
+      <h3>Clock {{clockOnState ? 'on to' : 'off from'}} the helpdesk</h3>
+    </div><!--/new-ticket-header-->
+    <div class="modal-body text-center clearfix">
+      <div class="col-sm-12" ng-if="clockOnState">
+        <label>
+          Please enter how long you are working at the helpdesk for.
+        </label>
+        <div class="input-group duration-entry col-sm-4">
+          <input name="duration"
+                 ng-model="duration.hours"
+                 class="form-control input-lg"
+                 type="number"
+                 placeholder="2"
+                 step=".25"
+                 min="0.25"
+                 max="8"
+                 required>
+          <span class="input-group-addon">hours</span>
+        </div>
+        <p class="text-danger" ng-show="sessionForm.duration.$error.number">
+          Please enter duration as a decimal number. For example, 1 hour and
+          15 minutes would be represented as <code>1.25</code> hours.
+        </p>
+        <p class="text-danger" ng-show="sessionForm.duration.$error.min">
+          You must work for at least fifteen minutes.
+        </p>
+        <p class="text-danger" ng-show="sessionForm.duration.$error.max">
+          You can't work more than eight hours.
+        </p>
+        <p class="help-block" ng-show="sessionForm.duration.$valid">
+          You will be automatically clocked off at <strong>{{automaticClockOffTime.format('hh:mm a')}}</strong>,
+          but you can manually clock off at an earlier time.
+        </p>
+      </div><!--/clock-on-state-->
+      <div class="col-sm-12" ng-if="clockOffState">
+        <label>
+          You finish working at the helpdesk in
+        </label>
+        <div class="time-left">
+          {{timeLeft.hours()}}h {{timeLeft.minutes()}}m {{timeLeft.seconds()}}s
+        </div>
+        <p class="help-block">
+          (That's at {{automaticClockOffTime.format('hh:mm a')}})
+        </p>
+      </div>
+    </div>
+    <div class="modal-footer text-right">
+      <button ng-click="$dismiss()" class="btn btn-primary">
```

```
+        Close
+      </button>
+      <button ng-disabled="!sessionForm.$valid" ng-click="clockOn()" class="btn btn-success" ng-if="clockOnState">
+        <i class="fa fa-clock-o"></i> <i class="fa fa-sign-in"></i> Clock On
+      </button>
+      <button ng-click="clockOff()" class="btn btn-danger" ng-if="clockOffState">
+        <i class="fa fa-clock-o"></i> <i class="fa fa-sign-out"></i> Clock Off Now
+      </button>
+    </div>
+  </form>
+</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, aler
+  $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
```

```
+    unitService.getUnit $scope.ticket.project.unit_id, false, false, (response) ->
+      $scope.ticket.unit = response
+      # If ticket was provided, we need to look up the task def now from the
+      # unit loaded if it has one
+      if not $scope.isNew and $scope.ticket.task_definition_id?
+        taskDef = $scope.ticket.unit.taskDef($scope.ticket.task_definition_id)
+        $scope.taskDefSelected taskDef
+
+  #
+  # Sends a HTTP request to open a ticket
+  #
+  $scope.openTicket = ->
+    openTicketCallback = (error, success) ->
+      if success
+        $modalInstance.close(success)
+        $rootScope.$broadcast('CurrentOpenTicket', success)
+        alertService.add("success", "Ticket created successfully.", 2000)
+      if error
+        alertService.add("danger", "Error: #{error.data.error}", 6000)
+    HelpdeskTicket.submitTicket(
+      $scope.ticket.project.project_id,
+      $scope.ticket.description,
+      $scope.ticket.task_definition?.id,
+      openTicketCallback
+    )
+
+  #
+  # Sends a request to resolve the ticket
+  #
+  $scope.resolveTicket = ->
+    # TODO: Resolved by whom?
+    resolveTicketCallback = (error, success) ->
+      if success
+        $modalInstance.close(success)
+        $rootScope.$broadcast('CurrentOpenTicket', null)
+        alertService.add("success", "Ticket resolved!", 2000)
+      if error
+        alertService.add("danger", "Error: #{error.data.error}", 6000)
+    $scope.ticket.resolve(resolveTicketCallback)
+
+  #
+  # Sends a request to close the ticket
+  #
+  $scope.closeTicket = ->
+    closeTicketCallback = (error, success) ->
+      if success
+        $modalInstance.close(success)
+        $rootScope.$broadcast('CurrentOpenTicket', null)
+        alertService.add("success", "Ticket closed.", 2000)
+      if error
+        alertService.add("danger", "Error: #{error.data.error}", 6000)
+    ConfirmationModal.show 'Close Ticket', 'Are you sure you want to close the ticket without resolving the issue?', ->
+      $scope.ticket.close(closeTicketCallback)
+
+  #
+  # Goes to the selected task
+  #
+  $scope.goToSelectedTask = $modalInstance.dismiss
+)
```

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

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

```
+.ticket-modal {
+  .unit-selection .dropdown {
+    width: 100%;
+    button {
+      text-align: right;
+    }
+  }
+  label {
+    margin-right: 1ex;
+  }
+  .unit-selection .dropdown-menu > li > a {
+    padding: 1em;
+  }
+}
```

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

```
@@ -0,0 +1,103 @@
+<div class="ticket-modal">
+  <form class="form-horizontal" role="form">
+    <div class="modal-header" ng-show="isNew">
+      <h3>Submit a Ticket</h3>
+      <p class="lead">
+        Having trouble with something? Get help from a tutor at the programming
+        helpdesk, located at ATC620.
+      </p>
+    </div><!--/new-ticket-header-->
+    <div class="modal-header" ng-hide="isNew">
+      <h3>Show Ticket</h3>
+    </div><!--/existing-ticket-header-->
+    <div class="modal-body">
+      <div class="form-group unit-selection">
+        <label class="col-sm-3 control-label">Unit</label>
+        <div class="col-sm-7">
+          <p class="form-control-static" ng-if="projects.length == 1 || !isNew">
+            <label class="label label-info small">{{ticket.project.unit_code}}</label>
+            <span class="unit-name">{{ticket.project.unit_name}}</span>
+          </p>
+          <div class="btn-group dropdown" ng-if="projects.length > 1 && isNew" dropdown is-open="status.isopen">
+            <button type="button" class="btn btn-default form-control dropdown-toggle" dropdown-toggle>
+              <span ng-show="ticket.project" class="pull-left">
+                <label class="label label-info small">{{ticket.project.unit_code}}</label>
+                <span class="unit-name">{{ticket.project.unit_name}}</span>
+              </span>
+              <span ng-hide="ticket.project" class="pull-left">
+                Select a Unit
+              </span>
+              <span class="caret"></span>
+            </button>
+            <ul class="dropdown-menu">
+              <li ng-repeat="project in projects">
+                <a ng-click="ticket.project = project">
+                  <label class="label label-info small">{{project.unit_code}}</label>
+                  <span class="unit-name">{{project.unit_name}}</span>
+                </a>
+              </li>
+            </ul>
+          </div>
+        </div>
+      </div><!--/unit-selection-->
+      <div class="form-group" ng-if="ticket.unit">
+        <label class="col-sm-3 control-label">Task</label>
+        <div class="col-sm-7" ng-if="isNew">
```

```
+          <task-definition-selector button-style="default" unit="ticket.unit" ng-model="ticket.task_definition" on-sele
+          <p class="help-block">
+            Select a task that you want help with, or leave blank if you have a
+            general question.
+          </p>
+        </div><!--/task-selector-if-new-->
+        <div class="col-sm-7" ng-if="!isNew">
+          <div ng-if="ticket.task_definition" class="selected-task-detail">
+            <label class="label label-info">{{ticket.task_definition.abbreviation}}</label>
+            {{ticket.task_definition.name}}
+          </div>
+          <div class="text-muted form-control-static" ng-hide="ticket.task_definition">
+            No task provided
+          </div>
+        </div>
+      </div><!--/task-selection-->
+      <div class="form-group">
+        <label class="col-sm-3 control-label">Description</label>
+        <div class="col-sm-7" ng-if="isNew">
+          <markdown-editor ng-model="ticket.description" placeholder="Enter description" height="100"></markdown-editor
+          <p class="help-block">
+            Give the tutor some context to your issue by providing a description
+            of your question. {{description}}
+          </p>
+        </div>
+        <div class="col-sm-7" ng-if="!isNew">
+          <div ng-show="ticket.description" class"form-control-static" ng-bind-html="ticket.description | markdown"></d
+          <p ng-hide="ticket.description" class="text-muted form-control-static">
+            No description provided
+          </p>
+        </div>
+      </div><!--/description-->
+      <div class="form-group" ng-if="!isNew">
+        <label class="col-sm-3 control-label">In Queue For</label>
+        <div class="col-sm-7 form-control-static">
+          {{ticket.lengthOfTimeOpen().humanize()}}
+          <small class="text-muted">
+            ({{ticket.created_at | date:'shortTime'}})
+          </small>
+        </div>
+      </div>
+    </div>
+    <div class="modal-footer text-right">
+      <button ng-click="$dismiss()" class="btn btn-primary">
+        Close
+      </button>
+      <button ng-disabled="ticket.project == null" ng-click="openTicket()" class="btn btn-success" ng-if="isNew">
+        <i class="fa fa-ticket"></i> <i class="fa fa-plus"></i> Submit Ticket
+      </button>
+      <button ng-click="closeTicket()" class="btn btn-danger" ng-if="!isNew">
+        <i class="fa fa-ticket"></i> <i class="fa fa-trash-o"></i> Close Ticket
+      </button>
+      <button ng-click="resolveTicket()" class="btn btn-success" ng-if="!isNew" if-role="Tutor Convenor Admin">
+        <i class="fa fa-ticket"></i> <i class="fa fa-check"></i> Resolve Ticket
+      </button>
+    </div><!--/footer-->
+  </form>
+</div>
```

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

```
@@ -0,0 +1,166 @@
+#
```

```
+# The programming helpdesk dashboard
+#
+angular.module('doubtfire.helpdesk.states.dashboard', [])
+.config((headerServiceProvider) ->
+  helpdeskDashboardState =
+    url: "/helpdesk"
+    views:
+      main:
+        controller: "HelpdeskDashboardCtrl"
+        templateUrl: "helpdesk/states/dashboard/dashboard.tpl.html"
+    data:
+      pageTitle: "_Programming Helpdesk_"
+      roleWhitelist: ['Student', 'Tutor', 'Convenor', 'Admin']
+  headerServiceProvider.state 'helpdesk', helpdeskDashboardState
+)
+.controller("HelpdeskDashboardCtrl", ($scope, $timeout, $rootScope, $interval, $state, HelpdeskStats, HelpdeskTicket, H
+  # Internal poll interval
+  pollInterval = null
+  # How long the duration is between each poll, in seconds
+  intervalDuration = 30
+
+  #
+  # Returns true if the stats are already being polled
+  #
+  isPolling = -> pollInterval isnt null
+
+  # Keep track of new stats|staff|tickets in these variable
+  $scope.data =
+    tickets: []
+    tutorsWorking: []
+
+  #
+  # This function is called when tickets have been updated
+  #
+  ticketsUpdated = (error, tickets) ->
+    # Only update what we need to
+    resolvedTickets   = _.differenceBy($scope.data.tickets, tickets, 'id')
+    newTickets        = _.differenceBy(tickets, $scope.data.tickets, 'id')
+    # Remove resolved tickets + merge new tickets
+    $scope.data.tickets =
+      _.chain($scope.data.tickets)
+      .filter((t) -> !_.find(resolvedTickets, { id: t.id })?)
+      .concat(newTickets)
+      .map((ticket, idx) ->
+        # Add ticket's position
+        ticket.position = idx + 1
+        ticket
+      )
+      .value()
+
+  #
+  # This function is called when stats have been updated
+  #
+  statsUpdated = (error, stats) ->
+    statusColours = {'Quiet': 'primary', 'Available': 'warning', 'BUSY': 'danger'}
+    statusEmojis = {'Quiet': '😃', 'Available': '😊', 'BUSY': '😱'}
+    statsTimes = _.keys(stats)
+    lastRecord = stats[_.last(statsTimes)]
+    avgWaitTime   = $scope.avgWaitTime   = lastRecord.average_wait_time_in_mins
+    numUnresolved = $scope.numUnresolved = lastRecord.number_of_unresolved_tickets
+    # TODO: Work out the right values
+    $scope.averageWaitTimeStatus =
+      if avgWaitTime < 6
```

```
+       'Quiet'
+     else if 6 <= avgWaitTime < 9
+       'Available'
+     else if avgWaitTime > 9
+       'BUSY'
+   numberUnresolvedStatus =
+     if numUnresolved < 6
+       'Quiet'
+     else if 6 <= numUnresolved < 9
+       'Available'
+     else if numUnresolved > 9
+       'BUSY'
+   $scope.averageWaitTimeEmoji = statusEmojis[$scope.averageWaitTimeStatus]
+   $scope.averageWaitTimeColor = statusColours[$scope.averageWaitTimeStatus]
+   $scope.numberUnresolvedColor = statusColours[numberUnresolvedStatus]
+   # Angular nvd3 accepts milliseconds not seconds for unix time
+   $scope.graphData = stats
+
+ #
+ # This function is called when staff have been updated
+ #
+ tutorsUpdated = (error, sessions) ->
+   # Same concept as ticketsUpdated
+   sessionsFinished = _.differenceBy($scope.data.tutorsWorking, sessions, 'id')
+   sessionsStarted  = _.differenceBy(sessions, $scope.data.tutorsWorking, 'id')
+   $scope.data.tutorsWorking =
+     _.chain($scope.data.tutorsWorking)
+     .filter((s) -> !_.find(sessionsFinished, { id: s.id })?)
+     .concat(sessionsStarted)
+     .map((s) ->
+       finishIn = s.timeUntilFinish()
+       s.minuteDisplay = Math.round(finishIn.minutes() + (finishIn.seconds() * 0.01))
+       s.hourDisplay = finishIn.hours()
+       if s.minuteDisplay == 60
+         s.minuteDisplay = 0
+         s.hourDisplay += 1
+       s.showHour = s.hourDisplay > 0
+       s
+     )
+     .value()
+   $scope.helpdeskClosed = $scope.data.tutorsWorking.length is 0
+   $scope.helpdeskOpen = not $scope.helpdeskClosed
+
+ #
+ # If clock on event, refresh immediately
+ #
+ $rootScope.$on 'CurrentWorkingSession', (event, session) ->
+   $scope.pollNow()
+
+ #
+ # Begins polling for stats
+ #
+ startPolling = (interval = intervalDuration) ->
+   return if isPolling()
+   pollForStats = ->
+     # We know if it's the first poll if we are not polling yet!
+     isFirstInterval = not isPolling()
+     # First poll we want to get the last hour's worth
+     subtractValue = if isFirstInterval then 1 * 60 * 60 else interval
+     useInterval   = if isFirstInterval then null else interval
+     from = moment().subtract(subtractValue, 'seconds').format()
+     to   = moment().format()
+     HelpdeskStats.get from, to, useInterval, (error, response) ->
```

```
+        $scope.lastUpdated = moment()
+        statsUpdated(error, response)
+    pollForTickets = ->
+      HelpdeskTicket.getUnresolvedTickets null, (error, response) ->
+        $scope.lastUpdated = moment()
+        ticketsUpdated(error, response)
+    pollForTutors = ->
+      HelpdeskSession.tutorsWorkingNow (error, response) ->
+        $scope.lastUpdated = moment()
+        tutorsUpdated(error, response)
+    pollFunction = ->
+      pollForTickets()
+      pollForTutors()
+      pollForStats()
+    $scope.pollNow = ->
+      pollFunction()
+    # Call poll at least once to start now
+    $scope.pollNow()
+    pollInterval = $interval pollFunction, interval * 1000 # milliseconds
+
+  #
+  # Stops helpdesk statistics from polling
+  #
+  stopPolling = ->
+    return unless isPolling()
+    $interval.cancel(pollInterval)
+    pollInterval = null
+
+
+  #
+  # Opens the session modal
+  #
+  $scope.openHelpdeskSessionModal = HelpdeskSessionModal.show
+
+  # When we load, start polling at intervalDuration seconds
+  startPolling(intervalDuration)
+  # When we unload, stop polling
+  $scope.$on '$destroy', stopPolling
+)
```

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

```
@@ -0,0 +1,105 @@
+// This mixin is used to hide on mobile devices specific rows
+@mixin hide-on-mobile {
+  @extend .hidden-xs;
+  @extend .hidden-sm;
+}
+
+#helpdesk-dashboard {
+  // Fix layout for medium
+  @media (min-width: $screen-md) {
+    height: calc(100vh - #{$navbar-height});
+    min-height: 600px;
+    display: flex;
+    flex-direction: column;
+    .row {
+      display: flex;
+    }
+    .status.row {
+      flex: 1;
+    }
+    .graph.row {
```

```
+      flex: 2;
+    }
+    .ticket.row {
+      flex: 1.5;
+    }
+  }
+
+  // Hide all rows on mobile except the tickets row
+  .row:not([class^="ticket"]) {
+    @include hide-on-mobile;
+  }
+  // Give rows some space between eachother
+  .row {
+    margin: 15px auto 0 auto;
+  }
+
+  // Help Header
+  h3.help {
+    @extend .text-muted;
+    margin-top: 0;
+    text-transform: uppercase;
+    font-weight: bold;
+    font-size: 1.2em;
+  }
+
+  // Status Row
+  .status.row {
+    font-size: 2em;
+    text-align: center;
+    .number {
+      display: block;
+      font-size: 4em;
+      line-height: 1;
+    }
+  }
+  // Tutor panel
+  .tutors-working {
+    align-self: center;
+    .list-group {
+      max-height: 300px;
+      overflow-y: scroll;
+    }
+    .list-group-item-heading {
+      font-size: 1.5em;
+      .name {
+        font-weight: bold;
+      }
+    }
+    .list-group-item-text {
+      margin: 0.7em auto 1em auto;
+      font-size: 1.2em;
+      display: flex;
+      flex-wrap: wrap;
+      .label:not(:last-of-type) {
+        margin-right: 1ex;
+        margin-bottom: 1ex;
+      }
+    }
+  }
+  // Ticket Row
+  .ticket.row {
+    justify-content: center;
+  }
```

```
+  // Helpdesk Closed
+  .helpdesk-closed {
+    height: 100%;
+    margin-top: 2em;
+    display: flex;
+    align-items: center;
+    text-align: center;
+    font-size: 1.5em;
+    .message {
+      margin: 0 auto;
+      button {
+        margin-top: 1em;
+      }
+      .title {
+        font-weight: bold;
+        display: block;
+        margin-top: 1em;
+        font-size: 1.5em;
+      }
+    }
+  }
+}
```

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

```
@@ -0,0 +1,74 @@
+<div id="helpdesk-dashboard" class="clearfix">
+  <div class="helpdesk-closed">
+    <div class="message" ng-if="helpdeskClosed">
+      <i class='fa fa-hourglass-end fa-4x'></i>
+      <span class="title"> Helpdesk Closed </span>
+      No tutors are working right now<br>
+      <button ng-click="openHelpdeskSessionModal()" if-role="Tutor Convenor Admin" class="btn btn-lg btn-success">
+        <i class="fa fa-clock-o"></i> <i class="fa fa-sign-in"></i> Clock On
+      </button>
+    </div>
+    <div class="message" ng-if="helpdeskClosed == null && helpdeskOpen == null">
+      <i class="fa fa-spinner fa-pulse fa-4x"></i>
+    </div>
+  </div><!--/helpdesk-closed-->
+  <div ng-if="helpdeskOpen">
+    <div class="status row">
+      <div class="people-in-queue col-sm-4 text-{{numberUnresolvedColor}}">
+        <span class="number">
+          {{numUnresolved}}
+        </span>
+        people in queue
+      </div><!--/people-in-queue-->
+      <div class="emoji-face col-sm-4">
+        <span class="number">{{averageWaitTimeEmoji}}</span> {{averageWaitTimeStatus}}
+      </div><!--/emoji-face-->
+      <div class="average-wait-time col-sm-4 text-{{averageWaitTimeColor}}">
+        <span class="number">
+          {{avgWaitTime}}m
+        </span>
+        average wait time
+      </div><!--/average-wait-time-->
+    </div><!--/status-row-->
+    <div class="graph row">
+      <div class="tutors-working col-sm-4">
+        <h3 class="help">
+          {{ tutorCount = data.tutorsWorking.length }}
+          tutor{{tutorCount > 1 ? 's' : ''}} working right now
```

```
+        </h3>
+        <div class="list-group">
+          <div ng-repeat="session in data.tutorsWorking" class="list-group-item">
+            <div class="list-group-item-heading">
+              <span class="name">
+                {{finishIn = session.timeUntilFinish(); session.user.name}}
+              </span><!--/tutor-name-->
+              <div class="finish-in pull-right text-muted pointer" tooltip="Finishes at {{session.clock_off_time | date
+                <span ng-if="session.showHour">
+                  {{session.hourDisplay}}h
+                </span>
+                {{session.minuteDisplay}}m
+              </div><!--/finish-in-->
+            </div>
+            <p class="list-group-item-text">
+              <label ng-repeat="unit in session.user.units_taught" class="label label-info pointer" tooltip="{{unit.nam
+                {{unit.code}}
+              </label>
+            </p><!--/unit-labels-->
+          </div>
+        </div><!--/tutors-working-list-->
+      </div><!--/tutors-working-->
+      <div class="graph col-sm-8">
+        <h3 class="help">
+          Tickets unresolved and wait times today
+        </h3>
+        <helpdesk-dashboard-timeline data="graphData" height="400">
+        </helpdesk-dashboard-timeline>
+      </div>
+    </div><!--/graph-row-->
+    <div class="ticket row">
+      <div class="ticket-wrapper col-sm-3" ng-repeat="ticket in data.tickets">
+        <helpdesk-ticket ticket="ticket"></helpdesk-ticket>
+      </div>
+    </div><!--/ticket-row-->
+  </div><!--/helpdesk-open-->
+</div>
```

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

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

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

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

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

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

```
+  replace: true
+  restrict: 'E'
+  templateUrl: 'visualisations/visualisation.tpl.html'
+  scope:
+    rawData: '=data'
+    height: '=?'
+  controller: ($scope, $timeout, taskService, projectService, Visualisation) ->
+    updateData = (rawData) ->
+      # Function to fix time as nvd3 needs a unix time in ms, not s
+      correctTime = (t) -> t * 1000
+      $scope.times = _.chain(rawData).keys().map(correctTime).value()
+      $scope.data = [
+        {
+          key: 'Average Wait Time',
+          bar: false,
+          values: _.map rawData, (data, time) ->
+            {
+              x: correctTime(time),
+              y: data.average_wait_time_in_mins
+            }
+        },
+        {
+          key: 'People In Queue',
+          bar: true,
+          values: _.map rawData, (data, time) ->
+            {
+              x: correctTime(time),
+              y: data.number_of_unresolved_tickets
+            }
+        }
+      ]
+      # Update forceY to be the max num
+      barMax =
+        _.chain(rawData)
+        .map((data) -> data.number_of_unresolved_tickets)
+        .max()
+      barMax = if barMax > 10 then barMax else 10
+      lineMax =
+        _.chain(rawData)
+        .map((data) -> data.average_wait_time_in_mins)
+        .max()
+      lineMax = if lineMax > 10 then lineMax else 10
+      $scope.options.chart.bars.forceY =
+      $scope.options.chart.bars2.forceY =
+        [0, barMax]
+      $scope.options.chart.lines.forceY =
+      $scope.options.chart.lines2.forceY =
+        [0, lineMax]
+      $timeout ->
+        $scope.api.refresh() if $scope.api?.refresh?
+
+    $scope.$watch (-> _.keys($scope.rawData)[0]), ->
+      return unless $scope.rawData?
+      updateData $scope.rawData
+
+    [$scope.options, $scope.config] = Visualisation 'linePlusBarChart', 'Helpdesk Dashboard Timeline', {
+      showLabels: no
+      margin:
+        top: 30,
+        right: 30,
+        bottom: 50,
+        left: 70
+      height: $scope.height
```

```
+      bars:
+        forceY: [0]
+      bars2:
+        forceY: [0]
+      lines:
+        forceY: [0]
+      lines2:
+        forceY: [0]
+      color: [
+        'darkred',
+        '#2ca02c'
+      ]
+      x: (d, i) -> i
+      xAxis:
+        axisLabel: 'Time'
+        tickFormat: (d) -> moment($scope.times[d]).format('hh:mm a')
+      x2Axis:
+        showMaxMin: true
+        tickFormat: (d) -> moment($scope.times[d]).format('hh:mm a')
+      y1Axis:
+        axisLabel: 'people waiting'
+        tickFormat: (d) -> d
+        axisLabelDistance: -12
+      y2Axis:
+        axisLabel: null
+        tickFormat: (d) -> "#{d}m"
+      y3Axis:
+        tickFormat: (d) -> 'l'
+      y4Axis:
+        tickFormat: (d) -> 'x'
+    }, {}
```

1 src/app/visualisations/visualisations.coffee View

```
@@ -9,6 +9,7 @@ angular.module('doubtfire.visualisations', [
   'doubtfire.visualisations.task-completion-box-plot'
   'doubtfire.visualisations.achievement-box-plot'
   'doubtfire.visualisations.achievement-custom-bar-chart'
+  'doubtfire.visualisations.helpdesk-dashboard-timeline'
 ])

 .factory('Visualisation', ($interval, analyticsService) ->
```

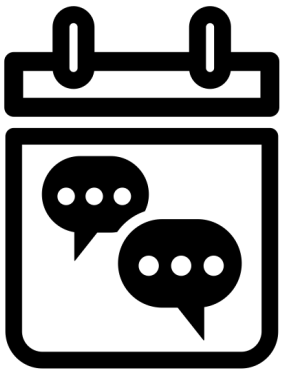
7 src/styles/common/alerts.scss View

```
@@ -49,3 +49,10 @@
   .alert.alert-warning i.fa:before { content: $fa-var-exclamation-triangle; }
   .alert.alert-success i.fa:before { content: $fa-var-check-circle; }
 }
+
+// Push down the alert past the header if it has content
+@media (min-width: $screen-sm) {
+  div[ui-view="header"]:not(:empty) ~ .alert-top {
+    top: 66px;
+  }
+}
```

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

```
@@ -0,0 +1,8 @@
+// Mobile responsive fixes
+@media (max-width: $screen-md) {
```

```
+  .modal-footer button {
+    width: 100%;
+    margin-bottom: 1em !important;
+    margin-left: 0 !important;
+  }
+}
```



Chapter 13

Meeting Minutes

Contents

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 Implementation[1]
2. Public Doubtfire Ticketing[2]

- We need to add a `Helpdesk` dropdown menu. Let's use `fa fa-support` for the icon:

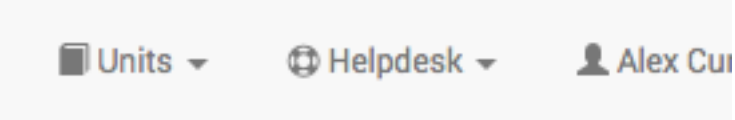


Figure 1: Helpdesk Dropdown

- 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

[1]See `https://trello.com/c/vKMN6hvN/47-rethink-doubtfire-backend-implementation`
[2]See `https://trello.com/c/qZJ2j4zQ/65-public-doubtfire-ticketing`

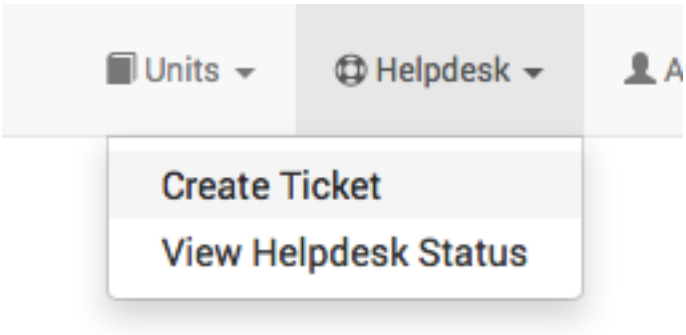


Figure 2: Create a ticket

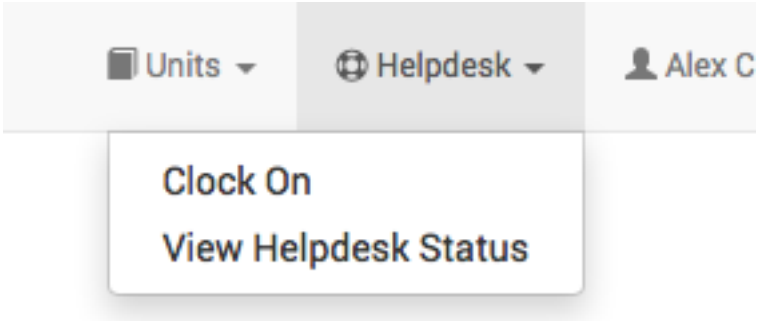


Figure 3: Clock on

- 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 defintion (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

[3]See `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

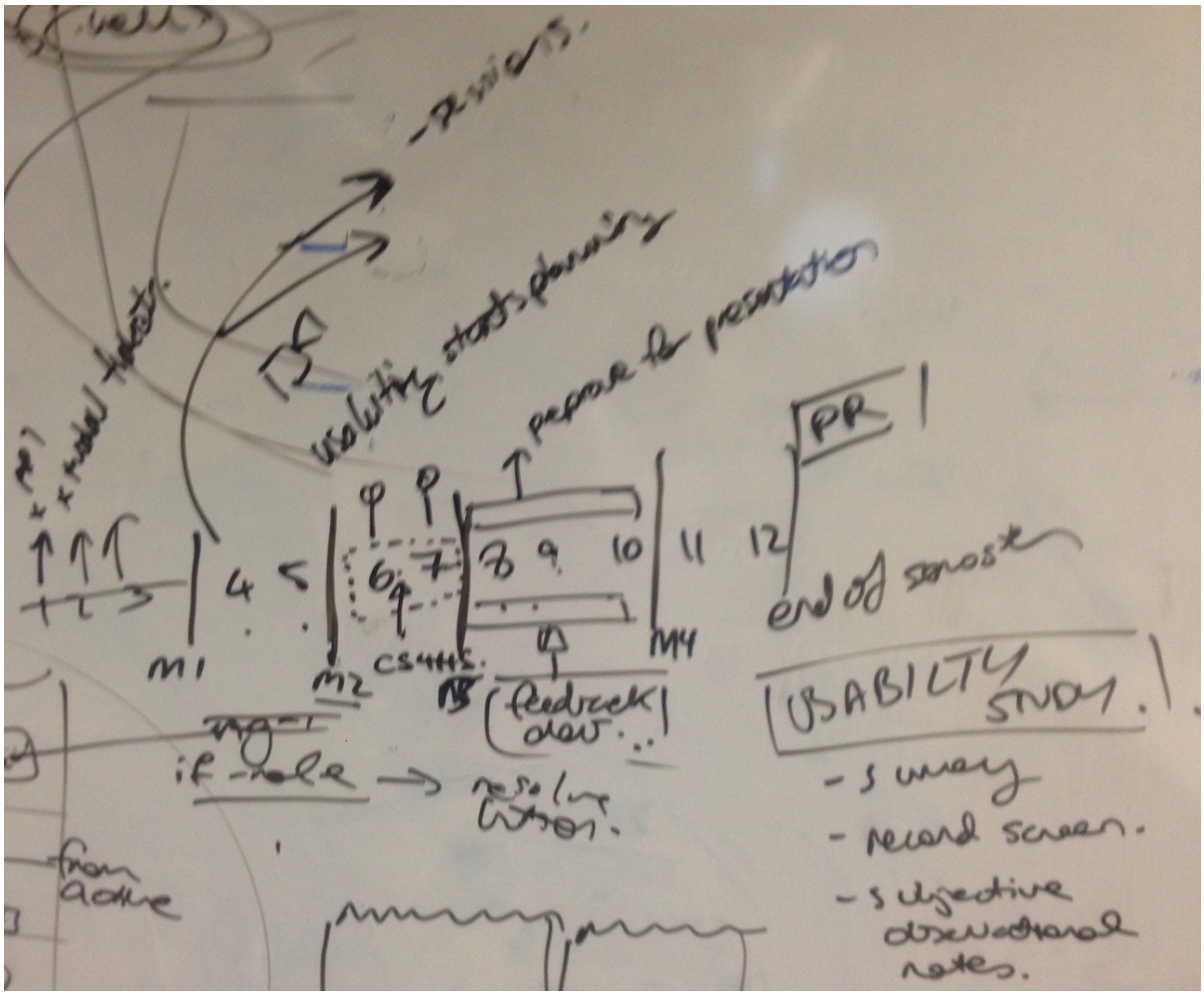


Figure 4: Sketch of timeline on whiteboard

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 if 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 testimonal 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

- Testimonals
    - 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