

BSc (Hons) Web Design & Development

Can a Web Interface improve the quality, reliability and accessibility of an Attendance Tracking system for an Organisation?

Written & Produced by Thomas Alderson

**SOUTHAMPTON SOLENT UNIVERSITY
School of Media Arts and Technology**

SOUTHAMPTON SOLENT UNIVERSITY
School of Media Arts and Technology

BSc (Hons) Web Design & Development

Academic Year 2015-2016

Thomas Alderson

Can a Web Interface improve the quality,
reliability and accessibility of an Attendance
Tracking system for an Organisation?

Tutor: M. Reid

April 2016

This report is submitted in partial fulfilment of the requirements
of Southampton Solent University for the degree of BSc (Hons)
Web Design & Development.

Acknowledgements

This report has been made possible through the effort of many supporting clients, the names of these acknowledgments are listed below along with the specific efforts that have been donated by the individual towards the completion of the project.

I sincerely thank Claire Alderson in her efforts of establishing contact between myself and the Brune Park Community College Attendance team to help gain vital knowledge key to the project research and development cycle.

I wish to express my sincere gratitude Lynn Dorey in her support of providing information regarding the existing staff attendance tracking system implemented in Brune Park Community College's and how the functionality feels for the current users of the system.

Special thanks to Martin Barret, Head of Information Technology at Brune Park Community College, for his help in reviewing the intended project definition and whether it is a suitable choice of project objective to fulfil and how the aims & objectives in the specification match to the current system installed at the school.

My gratitude goes to Abdullah Amsaeed for creating the Admin LTE 2 dashboard theme and allowing it to be made open-source to the public to use within any product for commercial, educational or personal use.

Most importantly, my sincere appreciations to Jomo Batola, for the ongoing support and supervision throughout the entire project development.

Abstract

This report intends to analyse, describe and explain the production segment of the project titled 'Can a Web Interface improve the quality, reliability and accessibility of an Attendance Tracking system for an Organisation?' along with the final outcome and conclusions to the project title. As the title is stated as a question, conclusions will be drawn using the correct analysis techniques and reflection on the project as a whole.

Contents

Acknowledgements	3
Abstract	4
List of Figures	14
List of Tables	15
Acronyms	16
Background & Introduction	19
Aims & Objectives	21
Draft Aims & Objectives	22
Aim	22
Objectives	22
Final/Current Aims & Objectives	23
Aims	23
Objectives	23
Option Analysis	26
RubyGems.....	26
User Limitation.....	27
Asset Pipeline.....	29

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Frontend Support	30
Customisable Courses.....	30
Limitation to Teachers	31
Personal Timetables	32
Policy Addition	32
Specification	33
Project Title	33
Project Aim & Objectives	33
Aim	33
Objectives	33
Application Details	34
Name	34
Colour Scheme.....	34
Project Interface Map Scope	34
Features	35
External Interface Requirements	35
User Interface	35
Hardware	35

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Software	36
Other Non-functional Requirements	36
Performance	36
Safety.....	36
Security.....	36
Methodologies	37
Project Cycle	37
Time Management	39
Idea Collection	40
Design	41
UX Documents	43
Backend Design	44
UX Documents	45
Low Fidelity.....	46
New Design Approach	46
Frontend Design	47
UX Documents	48

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Low Fidelity Designs	49
Mid Fidelity Designs.....	49
High Fidelity Design	50
Issues Arising from Implementation and Test	51
Implementation Phase	51
Necessary Software Dependencies	53
Software Creation.....	55
Frontend Integration	61
Issues During Implementation.....	62
Testing Phase	63
Device Testing.....	64
Criteria for Success Results.....	65
Evaluation & Conclusions	68
Overview.....	68
Aim	68
Objectives	69
Security	70
Responsive Design	71

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Sustainability & Support	71
Recommendations	73
Methodology	73
Technology.....	73
Design	74
References	75
Bibliography	82
Appendices	88
Appendix A: Importance of Software Testing	88
Appendix B: Original Project Proposal	89
Project Title	89
Introduction.....	89
Rational.....	89
Aim	89
Objectives	89
Intended Audience	90
Resources	90
Appendix C: Discussions Between Thomas Alderson and Claire Alderson via E-mail About Project Features	92

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Appendix E: Project Review Documentation.....	98
Appendix F: User Reference Guide	138
Appendix G: Project Definition Specification	140
Appendix H: Agile Methodology Cheat Sheet	143
Appendix I: Early Gantt Chart	151
Appendix J: Project Review Gantt Chart	153
Appendix K: Mind Map.....	155
Appendix L: Design Fidelity Grid	157
Appendix M: Backend Teacher Persona.....	158
Appendix N: Backend Design Hierarchical Task Analysis'	159
Appendix O: Backend Design User Journey.....	163
Appendix P: Low Fidelity Backend Designs	164
Design One	164
Design Two	165
Appendix Q: Admin LTE2 Github Repository	166
Appendix R: High Fidelity Backend Design Views	168
Login (/login).....	168
Dashboard (/dashboard)	169

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Messages Pages	171
Users (/users).....	176
Group Page (/groups/*).....	178
Timetable (/timetables/*)	180
Courses (/courses)	182
Appendix S: Frontend Design User Persona Example	184
Appendix T: Frontend Design Hierarchical Task Analysis'	185
Appendix U: Frontend Design User Journey.....	187
Appendix V: Low Fidelity Frontend Designs	188
Design One	188
Design Two	189
Design Three	190
Appendix W: Mid Fidelity Frontend Designs	191
Design One	191
Design Two	192
Appendix X: High Fidelity Designs.....	193
Appendix Y: w3 Website Implementation Plan	194
Appendix Z: Message Board Discussion of Using iMac as a Web Development Focused Computer	196

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Appendix AA: Ruby & Rails Command Line Cheatsheet	200
Appendix BB: ‘rails new’ Command Output.....	206
Appendix CC: ‘rails generate scaffold User’ Output.....	209
Appendix DD: Files Developed by Scaffold.....	210
Appendix EE: Application Gemfile	212
Appendix FF: Application Model Files	213
absence.rb	213
course.rb.....	213
group.rb	214
login_activity.rb	214
message.rb	214
note.rb	215
timetable.rb.....	215
timetable_entry.rb	216
user.rb	217
Appendix GG: User Controller	218
Appendix HH: Modal.js Source Code	219
Appendix II: Frontend Source Code	220

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Appendix JJ: Evidence of Device Testing 221

Appendix KK: Evidence of Timetabling Options 222

Personal Timetable View 222

Event View 224

User Profile Scheduling..... 226

Appendix LL: Article on ‘password_digest’..... 228

Figures

Figure 1: Agile Methodology	38
Figure 2: Gantt Chart Example	39
Figure 3: Low Fidelity Prototype Example.....	42
Figure 4: Mid Fidelity Prototype Example	43
Figure 5: High Fidelity Prototype Example.....	43
Figure 6: iMac Development Computer Specifications.....	52
Figure 7: Ruby on Rails Installation Search Results.....	53
Figure 8: iMac Command Line with Ruby and Rails Version	54
Figure 9: Routes.rb File Example.....	62
Figure 10: Removing Timetable ID from Groups.....	62
Figure 11: Adding TimetableID to Users.....	63
Figure 12: Evidence of Page Security	70

List of Tables

Table 1: SMART Definition	24
Table 2: Database Users Table.....	56
Table 3: Database Contents Table	57
Table 4: Database Enrolments Table.....	57
Table 5: Database Groups Table.....	57
Table 6: Database Login Activities Table	58
Table 7: Database Messages Table	58
Table 8: Database Timetables Table.....	58
Table 9: Database Timetable Entries Table	58
Table 10: Criteria for Success Results.....	66

Acronyms

Acronyms are used in order to abbreviate a collection of words to save a reader repeating themselves when reading, acronyms can be slang specific to the report if necessary, but throughout this report, abbreviations are the only technique used to shorten a collection of words for the reader.

1. CMS
 - a. **Expanded:** Content Management System
 - b. **Definition:** "A content management system, or CMS, is a web application designed to make it easy for non-technical users to add, edit and manage a website." (Plone n.d.)
2. CSS
 - a. **Expanded:** Cascading Stylesheets
 - b. **Definition:** "CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files." (w3Schools 2000)
3. CSS3
 - a. **Expanded:** Cascading Stylesheets 3
 - b. **Definition:** "CSS3 is the latest evolution of the Cascading Style Sheets language and aims at extending CSS2.1. It brings a lot of long-awaited novelties, like rounded corners, shadows, gradients, transitions or animations, as well as new layouts like multi-columns, flexible box or grid layouts." (Mozilla Developer Network 2016)
4. HTML
 - a. **Expanded:** Hypertext Markup Language
 - b. **Definition:** "The acronym HTML stands for HyperText Markup Language, the authoring language used to create pages on the World Wide Web. HTML is a set of codes or HTML tags that provide a web

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

browser with directions on how to structure a web page's information and features." (CuteRank 2010)

5. HTML5

- a. **Expanded:** Hypertext Markup Language 5
- b. **Definition:** "HTML5 was developed to solve compatibility problems that affect the current standard, HTML4. One of the biggest differences between HTML5 and previous versions of the standard is that older versions of HTML require proprietary plugins and APIs." (Rouse 2014)

6. IT

- a. **Expanded:** Information Technology
- b. **Definition:** "The technology involving the development, maintenance, and use of computer systems, software, and networks for the processing and distribution of data." (Merriam-Webster 2007)

7. JS

- a. **Expanded:** JavaScript
- b. **Definition:** "A widely used programming language that is embedded in most Web pages. Supported by all Web browsers, it enables interactive functions to be added to Web pages, which are otherwise static. JavaScript evolved from Netscape's LiveScript language." (PCMag 2013)

8. OS

- a. **Expanded:** Open Source
- b. **Definition:** "When a software program is open source, it means the program's source code is freely available to the public. Unlike commercial software, open source programs can be modified and distributed by anyone and are often developed as a community rather than by a single organization." (Tech Terms 2008)

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

9. Rails

- a. **Expanded:** Ruby on Rails
- b. **Definition:** "Rails combines the Ruby programming language with HTML, CSS, and JavaScript to create a web application that runs on a web server." (Kehoe 2013)

10. UI

- a. **Expanded:** User Interface
- b. **Definition:** "...it is possible to define a User Interface as a set of interaction elements which let a user carry out a task in a specific context." (Gallud, Tesoriero and Penichet 2011)

11. UX

- a. **Expanded:** User Experience
- b. **Definition:** ""User experience" encompasses all aspects of the end-user's interaction with the company, its services, and its products." (Norman and Neilsen 2013)

Background & Introduction

This report is intended to outline and describe the progression of the project titled 'Can a Web Interface improve the quality, reliability and accessibility of an Attendance Tracking system for an Organisation?' an investigation and question that is expected to be answered/solved through the development of an attendance tracking application that can be solely accessed through any web browser available to a user connected to a local network of an establishment.

To fully bring the project to life and to collect relevant data to the use of the application and whether or not it is successful, a scenario was devised, a school, known as Brune Park Community College was contacted by the project developer to discuss how their current attendance tracking system for their staff is working. The results from the questioning returned negative, according to **Alderson (2015)**, deputy head of Year 11 for the school, the current system needs upgrading and can regularly cause confusion for the current staff.

Using the information provided by staff members of the school, the scenario of inventing a web application to test in the school came to life, the help render a problem created by the current system non-existent. The application was to be developed as if soon to be deployed on the school's private network, over inventing an application for the sake of invention, with a scenario, the plan becomes more cemented/real.

A full understanding of the importance of testing and why real-life simulations/implementations can benefit the testing phase is specified in Appendix A.

The intentions of the application are for it to be ran using Ruby on Rails, a web development framework created to help build web based applications using the programming language known as Ruby. A full explanation of what Ruby on Rails is has been given by Hansson...

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

"Rails is a web application development framework written in the Ruby language. It is designed to make programming web applications easier by making assumptions about what every developer needs to get started. It allows you to write less code while accomplishing more than many other languages and frameworks. Experienced Rails developers also report that it makes web application development more fun." (Hansson 2009)

Rails was chosen as a suitable option to allow for the developer to broaden their knowledge to understand the wider availability of programming languages to create highly advanced web applications/websites that are available. Although the developer has experience within PHP, Ruby can be a helpful language to understand and can be applied to future projects outside of the web if needed.

Stated by Bien, the programming language known as Ruby is growing in high popularity and demand by employers...

"The TIOBE Programming Community Index lists Ruby as the 13th most popular programming language as of the time of writing this article. Although popularity is a good thing, what's better is demand. Currently, Ruby is experiencing incredible growth in demand." (Bien 2013)

To fully understand the expected outcome and what the intentions of the project were to begin with in much more detail, please see Appendix B for the Project Proposal.

Aims & Objectives

Specifying the aims and objectives are beneficial to help outline the definitive expectations that the final product must meet, without setting aims and objectives, the possibility of measuring the level of success within the project would not exist. According to Solent University (n.d.), aims are needed in any project to specify the means of intent in statement format, whereas objectives are measurable units that can be used to identify the project's success using a unit fashion or steps, together they help provide a strong positive unit of measurement to support why the project is a success or failure.

The development of the aims & objectives have been a forever growing and forming process, although primarily there was a primary aim set with a number of objectives, these were simply a baseline to work from – as the project evolved and further investigation of Brune Park Community College's existing installation conditions began to be understood, it was clear that more concise aim's and objectives would need to be stated, built upon the bases of the existing aims and objectives the project research was being adhered to.

A fantastic explanation of the link between writing correct aims for a task and the task itself, provided by Charities Evaluation Services...

"Your mission or overall aim is likely to be too broad a statement to allow you to plan your work in detail, or to provide guidance on what you might monitor and evaluate. Specific aims – statements about different aspects of your main purpose – will allow you to think clearly about what you want to achieve."

(Charities Evaluation Services n.d.)

Throughout this section of the report, the aims and objectives will be displayed in evolutionary order to allow readers to fully understand the changes that the project has been put through.

Draft Aims & Objectives

At the very start of the project, its birth as it can be called, developments were made into discovering the exact expectations of the outcome, and to ensure that the correct research had begun, trial aims and objectives were set to allow for 'skeletal' research to begin – such as the expected portrayal of the new application in terms of programming language and structure for a site-map, using existing examples.

Aim

There is currently one specific Aim that the project needs to cater to, with objectives supporting the following statement...

"To build a personal programming and server development skills to manufacture an Attendance Tracking system/software that is capable of use in an organisation environment."

Objectives

Proceeding forth from the statement specified above that summarises the basic Aim of the whole project, we have the objectives following closely behind, these need to also be met, but by using the objectives, we have a deeper understanding of what the product needs to be able to achieve upon completion.

The aim of this project is to address the presented problem by creating a web-based interface that is...

- Attendance Tracking System with the ability to supply each user with a personal timetable, attendance record and file management system.
- Easily accessible and usable across multiple platforms that, holds strategic reliability through the use of a sturdy, secure database structure.
- Fluid in terms of design to allow for easy changes through CSS/SASS/SCSS if colour changes or content changes need to be implemented.
- Responsive to any device it is accessed by, caters to all screen resolutions.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

These 'Early Stages' aims and objectives that were created before the full understanding of the expectations of the Brune Park Community College were not designed entirely to help the implementation and testing phase that would come later in the development of the project, as they do not give any possible unit of measurement to the development team on whether the created product has fulfilled its expectations or not.

The aims and objectives were updated as soon as possible throughout the research stage of the project to be a much more fit for purpose generated check-list to ensure that the final product achieves the desired outcome – this was accomplished by combining the research made into existing systems and the opinion of existing Brune Park Community College staff on the features they would like implemented and how they can go about accessing these.

Please see Appendix C for a fully detailed discussion between Claire Alderson and the project developer all about the features that are desired in the final build.

Final/Current Aims & Objectives

These are the final decided aims & objectives after the discussion had taken place between Clare Alderson and Thomas Alderson, outlining the now measurable objectives and final aim that the project was to be based around.

Aims

To build personal programming and server development skills to manufacture an Attendance Tracking software that is capable of use in an educational organisation environment for tracking and timetabling staff's personal daily schedules.

Objectives

- Secured accessibility, using login and high level encrypted passwords.
- Accessible Web Interface, scalable to be accessed on any device for portable usage when connected to network controlling application.
- Sustainable on a system with a maximum of 200 user accounts and regular connections.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

- Primary support for top five most used web browsers, judging by global web statistics.

The aims and objectives were also created in the format known as SMART, the SMART format of setting up business/project goals is laid out as an abbreviation of the following segments...

Table 1: SMART Definition

S <i>Specific</i>	"A specific goal has a much greater chance of being accomplished than a general goal." (Top Achievement 1999)
M <i>Measurable</i>	"Measurable goals means that you identify exactly what it is you will see, hear and feel when you reach your goal. It means breaking your goal down into measurable elements. You'll need concrete evidence." (YourCoach 2010)
A <i>Achievable</i>	"There is little point in setting a goal that is either too difficult to achieve or beyond your capabilities, as this will only serve to demotivate you and destroy your self-confidence. The importance of being able to accomplish a goal is equally vital when you are setting goals for others, as it is for yourself." (Free Management EBooks 2014)
R <i>Realistic</i>	"Is the goal relevant? Do you actually need to increase the activation rate? The best way to check relevancy of the goal is to think about the reason why you've established the goal in the first place. If the goal relates to the reason, then you're in good shape." (Walczak 2015)
T <i>Time</i>	"...objectives just don't get done when there's no time frame tied to the goal-setting process." (Zahorsky 2016)

To see the SMART Format of the specified final objectives, please see Appendix D.

The aims and objectives slowly developed as the project planning stages were implemented, due to the constant cycle of regular reviews made to the project as it was evolving. The project intentions certainly became cemented during the review period at which a write up was created to review each detail of the project and how it has been developed in contrast to expectations, if there were any.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Please see Appendix E containing the Project Review documentation, detailing the changes in the project, along with the progress after 4 months.

Option Analysis

Within the project production cycle, there are a number of aspects/features that can be developed and changed in order to allow the product to fit to the client specification that has been outlined in the research and development stages of the project cycle. "Typically an option analysis is a strategic report that provides quantitative and qualitative details of each option, associate risk factors and assesses impact in the current system for project stakeholders. The findings can be used as an input to complete/adjust Enterprise Architecture future state system definitions." (Villanueva 2011)

The project has limited, yet flexible boundaries in terms of the technology that it relies on to function. Within Ruby on Rails, there are many features available by using Ruby gems, these are dependencies that can be installed within a Ruby on Rails application to allow the application developers to access the new features by calling specific functions or methods within the HTML view pages that are generated at the beginning of the project.

RubyGems

Through the help of RubyGems (this comes installed on Ruby on Rails), Package Libraries are able to be installed on the application via the simple editing of a file known as a Gemfile, located in the root directory of the application. Adding a gem is achieved by calling 'gem '[GEM NAME HERE]' and can then be installed to the application by calling 'bundle install' this will use the gem to download all of the necessary assets that the specific feature that the user wants installed which can then be called on using the JavaScript functions or styles that the gem has installed for the user.

A fully fledged extensive explanation on what RubyGems is and what exactly gems are has been given by the development team at RubyGems within the introductory documentation...

"The RubyGems software allows you to easily download, install, and use ruby software packages on your system. The software package is called a "gem"

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

and contains a package Ruby application or library. Gems can be used to extend or modify functionality in Ruby applications. Commonly they're used to distribute reusable functionality that is shared with other Rubyists for use in their applications and libraries. Some gems provide command line utilities to help automate tasks and speed up your work." (RubyGems 2011)

The reason for explaining about what RubyGem is along with it's children known as gems is due to the rails application being ran in development mode even when it is fully implemented to an organisation's network, this allows for regular maintenance to be made to the application and extra features to be installed, thanks to communities such as GitHub (a repository website that is used to host projects for teams or features for applications, as explained by Brown, 2014), hence the choice for Ruby on Rails as the development technology.

User Limitation

Scoping the amount of users that the system can hold is beneficial towards the individual organisation that the system is being installed within. Due to high network access, a lot of power can be taken from the use of other applications, which can cause drainage within the application's memory, resulting in a slow performing application; this means that a server with a high specification must be able to support the application's functionality, based on the amount of users the system is planning to hold.

As Brune Park Community School is a small secondary school facility and houses up to near 150 staff members, whether they be teachers, receptionists or support staff, the server that controls the data management of the school is high-end quality, with a number of servers interconnected capable of a minimum 2 gigabyte's of RAM, with processors that can handle the control of incoming connections and requests to the server as each web page is loaded, produced by Intel.

The more users that the system holds, the larger amount of data there is to be stored within the server and a lot of strain on the server as there will be a large collection of connections being sent inward and outward bound, causing a vast amount of work to be carried out by the network – which can result in a slow

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

experience for the user, this is where the reference table comes into play based on the technological capability that the organisation is using to function the application, the table will return an amount of users suitable for the setup.

Please see Appendix F for the User Reference Guide.

Although, dependant on the type of server that is being used can vary on the size of the database available on the server itself, if an organisation would rather use a database built into the server, if not a database is supplied with Ruby on Rails by default and runs off of this and is purely file constructed and is a Postgre SQL database format.

As explained by Microsoft, server ran databases can have their limitations...

"The SQL engine supports 1 CPU, 1 GB RAM, and a 4 GB database size. This mechanism permits easy differentiation from other SQL Server 2005 editions by having well defined cut-off points. Otherwise, there is no workload throttle and the engine performs as in other editions. There is no hard-coded limit to the number of users that can attach to SQL Server Express but their CPU and memory limits impose practical limits on the number of users that can achieve acceptable response times from a SQL Server Express database."

(Microsoft 2011)

...and here is a short description on what a PostgreSQL database is, the database that is stocked within Ruby on Rails and will be used to host the data for Brune Park Community School by default selection of the application, due to being developed using Ruby on Rails and cost efficiency matters...

"PostgreSQL is a powerful, open source object-relational database system. It has more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness." (The PostgreSQL Global Development Group 2004)

Asset Pipeline

The asset pipeline is a feature that comes stocked with the Ruby on Rails application upon installation, although the layout of the pipeline is very limited and this certainly needed changing to cater towards the specific users that will be using the application, in this case; the candidate being Brune Park Community School. Due to the primary usage in this case being a school, the opportunity to allow maintenance to be openly available to an IT team to make changes or add features if they would like to.

The benefits of the asset pipeline rely solely on the fact that Ruby on Rails sets the standard folder structure for the assets that can be included by the maintainer of the application, if new content is desired, then the images needed can be dropped into the asset pipeline folders, located at '/app/assets' within the application's root destination.

The asset pipeline comes stocked with three folders, labelled 'images', 'javascript' and another called 'stylesheets' – although these are not accessed by the application by default. Ruby on Rails compiles all CSS and JS files together into one to save load time, meaning the compilation list is identified in the two files labelled 'application.js' and 'application.css' – these options had to be changed by adding the necessary folders to the pipeline list for the specific files.

A better explanation is specified in the documentation for the Ruby on Rails' pipeline...

"Sprockets concatenates all JavaScript files into one master .js file and all CSS files into one master .css file. As you'll learn later in this guide, you can customize this strategy to group files any way you like. In production, Rails inserts an MD5 fingerprint into each filename so that the file is cached by the web browser. You can invalidate the cache by altering this fingerprint, which happens automatically whenever you change the file contents." (RailsGuides 2011)

Frontend Support

As of the original plan, frontend support was not a feature that was expected to be implemented in the early stages or even within the stages of production, until the detailed production of the courses/classes had begun; this segment of production was expected to be non-existent and left purely to the client, the only issue with leaving tasks like this purely to the client without any reference is it can become confusing with the structure of Ruby on Rails directories; therefore having an example front-end design in place will allow future users to view an existing product and replicate it to their own unique design(s).

By connecting the front end website to the backend control panel allows the navigation to update with any new courses that are added to the database, meaning the frontend website does not need to be touched by anyone, it can be all developed and maintained through the back-end control centre, accessible to all staff members with an account.

At a simple level, Moveable Online describe a key benefit of using a CMS-like system to control content that every control management system boils down to...

"Not all users have the same comfort level with technology, but the basic CMS functions of writing and publishing content, and slightly more advanced ones of adding media are usually easy for everyone to grasp. In fact, anyone who can use word-processing software can use a CMS for the basic functions – so you don't even have to spend much time on training." (Moveable Online 2013)

Customisable Courses

Tying into the frontend support features that intend to be implemented, the product will act as a part-time content management system, as to allow staff members to change the content of their courses on demand, by doing this; the frontend design/description of the course changes, allowing all users accessing the organisation's website to witness the new content as soon as it is available.

Limitation to Teachers

The original design within the project was to allow the product to be a local portal for all members of an organisation, and as Brune Park Community School is full of students and faculty, the original intention was to allow the product to be fit for purpose in both parties; although this feature changed during the Project Review era, at which students were decided to be subtracted from the application due to lack of speed becoming an ongoing issue.

Speed is an important aspect to any application, and with a large set of users, speed begins to get sacrificed for support across a multiple type of stakeholder. A perfect example of how this can affect users of the application in a negative light is presented via MightyText's support message board, an application used to send text messages through android phones via your desktop computer.

"I enjoy upgrades and new features. However, I also use mighty text to do things fast. I do not enjoy the new loading screen that slows down the initial load time. Also - now after some update that you did switching between messages is not lightning fast anymore. I have to wait for it to load and spin a little bit. Give us the option to have just a plain vanilla experience please where super fast load times of everything is the number 1 priority. I don't care about color, I care that when I switch to another persons message it takes 2 seconds before I can type to them! Thanks!" (j3ffrey 2014)

As determined by the quotation above, adding new users (data) or in this user's case, features (also data) can result in speed being lost, hence the fighting argument that a plain vanilla experience is more important to a user because it allows for beneficial speed ratings to remain consistent.

The result of the speed loss in the prototype application was much the same, although excessive users caused the slow speed due to the amount of bandwidth being used at once, this can be easily averted by a higher server with more bandwidth but can result in a higher cost for the organisation with the application installed; thus the application took a feature change of being strictly for teachers rather than an entire school; although the application can still be used by entire

organisations, based purely on their services they offer and how many user groups they need to support within the system.

Personal Timetables

Due to the original plans of the product expectations supporting multiple users, the Timetable feature was expected to be a group-dependant object, meaning a timetable would belong to a group, which each user would then inherit and be expected to follow; this is simply due to the system relying upon tutor groups to sort the scheduling system.

The structure of the timetabling system was changed to be personal to each individual user, rather than being group-orientated, due to the change from students and teachers to simply teachers, this also helps the application be flexible for a multitude of organisations, as opposed to being specific to a niche market.

Policy Addition

The effect of the policy addition regarding a new system ties into the school or organisation that is using the application's legislation, users must understand that the system abides to the same laws and principles of all current systems within the organisation; this change is not huge and does not directly influence any changes on the project's point of view, although from the organisation's point of view that is in use of the application, it can be an issue.

This problem is a candidate solution because the security embedded within Ruby on Rails is enough to cover user protection in terms of data, along with offering beneficial misuse act with backup options within Ruby on Rails should there be a problem regarding Computer Misuse.

Specification

Throughout the project, the specification has been regularly updated to reflect the changes in the development decisions based on the clients needed and regulations, dependant on either technology or simple personal choice from the client. At the point of the Project Review, conducted within January 2016, the specification remained consistent and showed no sign of needing to be adapted for any new purpose.

Please see Appendix G for the Project Definition Specification, this was the first specification defined.

Project Title

Can a web interface improve the quality, reliability and accessibility of an attendance tracking system?

Project Aim & Objectives

The projects original aim & objectives were laid out in the project proposal, and after much deliberation with the aim & objectives as the project production has continued, they need to be adapted and developed to match the intended outcome of the project's final expectations.

Aim

To build personal programming and server development skills to manufacture an Attendance Tracking software that is capable of use in an educational organisation environment for tracking and timetabling staff's personal daily schedules.

Objectives

The expected objectives have changed since the original that were detailed in the previous project specification and proposal. The, now updated, project outcome objectives are detailed below...

- Secured accessibility, using login and high level encrypted passwords.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

- Accessible Web Interface, scalable to be accessed on any device for portable usage when connected to network controlling application.
- Sustainable on a system with a maximum of 200 user accounts and regular connections.
- Primary support for top five most used web browsers, judging by global web statistics.

Application Details

These details are simply basic levels of decisions within the details of the application, this way the application details can be referred to at any point in the current documentation.

Name

The name of the application is Hubbal, a combination of the words Hub and Portal, this is the name of the product, the organisation that installs Hubbal can be named anything.

Colour Scheme

The decided colour scheme at a basic level will be grey/blue, these are simply because of the developers favourable aspect of these two colours, although backup evidence will be represented for these colours within the design sector of the application.

Project Interface Map Scope

The scope is a short, crisp list presenting the mapped elements that the project will cover when in use. This scope list will be catered specifically towards Brune Park Community School, as this is the educational organisation that the testing and implementation will be used for.

- Login
- Main Application
- Dashboard
- Messages

- Timetable
- Users
- Group
 - List of groups to go here.
- Classes/Courses
- Logout

Specifying this layout allows a coverage to be measured of navigation and hopefully will also ensure that the application contains all of the necessary features intended through the original design discussed in the proposal. Giving a single value of measurement for completion to take into account.

Features

The primary function of the final creation will help educational business and organisations track their attendance records through the use of registration areas, supported by rankings and groupings to differentiate users from each other. The system is primarily intended to allow staff to edit, create and manage files and learning resources that can be accessed publicly, this is to allow staff to offer places for students to view their course information if needed.

External Interface Requirements

User Interface

- Follow conventions of standard Dashboard styles & templates.
- Left hand layout with the ability to be hidden.
- Responsive design to support multiple devices.
- Neutral colour design to suit all types of users.
- Typical horizontal navigation to contain User-specific information.

Hardware

- Moderately powered device to contain Ruby on Rails server, 2GB RAM and memory of at least 4GB.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

- Screen needed to control server and make changes of front-end if needed.
- Mouse and keyboard are also needed.
- Networking abilities to connect to the internet.

Software

- Administration Privileges to control installation of server.
- Web Development Software to make front-end changes if necessary.
- Ability to run background processes to support Ruby on Rails.

Other Non-functional Requirements

Performance

- The speed of the system is based entirely on the device that the software is performing on. This is why a baseline has been specified above in the Hardware requirements.
- Operating system has no effect on performance, providing that there is a sufficient device for the software to function on.

Safety

- Health and safety terms are based purely on the policies written by the business using the product.
- Usual safety precautions must be taken such as Data Protection and filter for curse or unacceptable language.

Security

- Data Protection Act Compliance.
- Computer Misuse Act Compliance.
- Copyright Designs and Patents Act Compliance.

Methodologies

The project had a number of methodologies that were followed to bring the final design to life, due to the high amount of production that the project had, along with the number of techniques that were used collectively to create specific aspects of the project, there were a multitude of methodologies that were used for each task at hand. The importance of a methodology for each task is extremely important in any project, it can allow the stakeholders to be confident in the expenditures that are being made, it can also help support organisation in the project development cycle, as discussed by Rangahau, 2008.

The tasks that are going to be discussed, along with the methodology chosen for each one, are the following...

1. Project Cycle
 - a. This is the main project cycle at which rate all tasks are to be completed.
2. Time Management
 - a. The technique of time management used to plan and log all specific tasks as to when they will be and were completed.
3. Idea Collection
 - a. Methodology used to collect ideas for the project ideology.
4. Design
 - a. The choice of technique(s) on how to design the basic and highly detailed design of a feature or theme in the project.

Each segment of the project life cycle was a ‘small, minified’ project in itself, allowing for each section to be dissected and analysed and discussed on whether the chosen methodology per task was the correct approach or not.

Project Cycle

The chosen methodology that guided the project from planning to production and finally implementation, was the Agile methodology, with this being an independent

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

project with only one team member, the usage of Agile was a surprise to assessors of the project as it was ongoing.

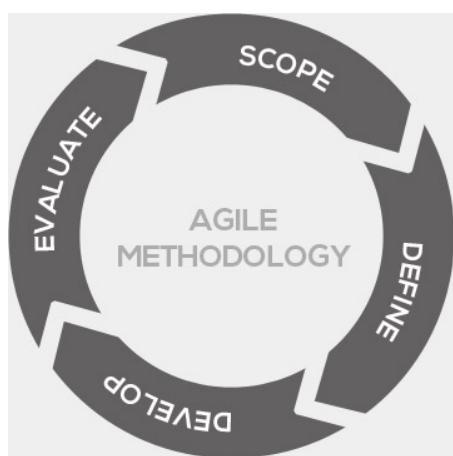


Figure 1: Agile Methodology

(Net Solutions 2012)

Due to the project being a University assignment as a project, the use of Agile was beneficial within the production cycle of the project; as explained by Kelly Waters, an Agile methodology professional, "A key principle of agile development is that testing is integrated throughout the lifecycle, enabling regular inspection of the working product as it develops. This allows the product owner to make adjustments if necessary and gives the product team early sight of any quality issues." (Waters 2007)

For a cheat sheet that was referenced throughout the project development all about the Agile methodology, please see Appendix H.

Although ongoing testing and implementation was a huge beneficial gain towards the project development, especially with the development cycle being performed entirely through Ruby on Rails, this was not the only advantage using the Agile cycle. The usage of rapidly developing features via regular sprints, used within Scrum methodology, a task approach built into Agile methodologies, ensure that the developers time could also be paid attention to other projects being performed in their own personal career – allowing knowledge to be gained from third party projects, outside of this one.

"During a sprint, you conduct constant inspections to assess progress toward the sprint goal, and consequentially, toward the release goal." (Layton 2013) as discussed by Layton, a sprint is a step in the project that brings the project towards its final goal. A regular sprint lasted consecutively of twenty-five minutes and a feature was to be completed to as higher standard as possible in terms of functionality, the outer design of the feature was changed after the sprint was completed, as sprints were only catered towards production in terms of functionality and not design.

Time Management

Time management is an important aspect of the project and allows for all of the tasks to be planned in preparation before they are performed, as specified within the Project Definition, seen at Appendix G. As discussed by Forsyth, Time is an important aspect that must be regulated; "Whatever job you do, if you are in a management or executive role, you will utilise a number of resources. People, money, materials – all are important. In any particular job, one resource may predominate. But there is one resource we all have in common: time. And time is a hard taskmaster." (Forsyth 2010)

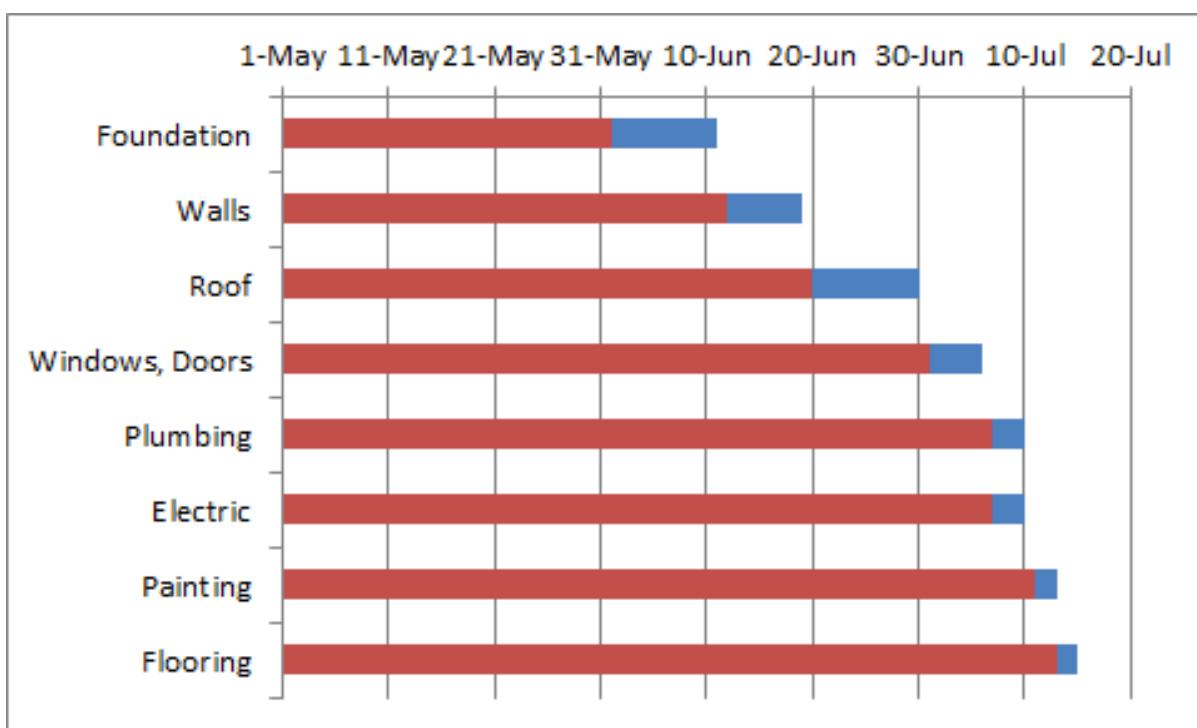


Figure 2: Gantt Chart Example

(Excel Easy, n.d.)

As displayed in the figure above, having a time management methodology/technique set in place ensures that all tasks are logged and are intended to be completed by a specific time frame. The chosen methodology of time management was a Gantt chart, used to give a full illustrative visual representation of the project development, invented in 1917, a Gantt chart was a new development in time management techniques.

Explanation of how a Gantt chart came to be, by Polakov and Trabold...

"Mr. Gantt concentrated his attention on the development of a method of charting which would show a comparison between performance and promises. Several years previous to this time, he had used a chart on which the work for machines was "laid out" according to the time required to do it."

(Clark 1923)

Building on from the history and the production of the Gantt chart and how it came to be, the uses of the Gantt chart offer a multitude of benefits for any project, with the core benefit being the visual aspect, as conveyed by Martinelli and Milosevic, 2016.

The Gantt chart evolved rapidly throughout the project cycle, adding, subtracting and changing the time length of tasks became an everyday occurrence, please see Appendix I for the earliest rendition of the project Gantt chart. As the project progressed, the time management tool needed to be more in-depth than a simple overview, and thus was rectified to display the length of a task purely based on the amount of minutes the specific task would take to complete, before full development period of the project was to begin.

Please see Appendix J for the Gantt chart within the Project Review stage.

Each task was performed on time according to the Gantt chart, even along with the exceptions made, explained in the Option Analysis to suit the candidate's desires/expectations or issues that had been raised once production had begun.

Idea Collection

The collection of ideas in any project is a task that must be completed, through the use of creating specific documents to help contain the starting ideas that could possibly be used within the application allow for a visual representation to be reached to determine which ideas are and are not beneficial to the project. There were a multitude of technologies that could be used to methodise the ideas at the

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

planning stages of the project, but to illustrate the ideas, a Mind Map was developed.

Athanasiou, Debas and Darzi's published works give a strong definition as to how useful a mind map can be in illustrating project ideas...

"One of the ways to organise thoughts is the use of mind maps. These are diagrammatic presentations of ideas and words that are interconnected according to the flow of information and the use of mind maps can help separate and clarify the thought processes for creating presentations or manuscripts." (Athanasou, Debas and Darzi 2010)

Please see Appendix K for the Mind Map produced during the Project Definition phase.

Design

The design process used a specific methodology that is a standard basis of every project within the design stages, the three specific design steps that were followed to achieve the designs created were the following...

- Low Fidelity
- Mid Fidelity
- High Fidelity

The three design phases have been defined by Pacheco, an expert in user experience and user interface development; "Low fidelity prototypes consist of sticky notes and sketches, which is great for high-level brainstorming and collaboration. Mid fidelity prototypes are often called wireframes. High fidelity prototypes almost represent the finished product." (Pacheco 2014)

These design phases were used in the design segment of the project cycle and will be defined in more detail in the design section of this report.

Design

The design aspects of the application were drawn up and brought to fruition through following the methodology discussed in the previous section of following a low fidelity, mid fidelity and high fidelity step process to help mimic the design as a prototype; the technology used at each step varied and in some cases of the project, the low fidelity, mid fidelity and high fidelity technique was completely abandoned to make way for a different approach entirely.

Please see Appendix L for an explanation grid detailing the level of fidelity and it's attributes.

Egger, an MSc in Ergonomic and HCI, discusses the benefits of using low fidelity designs in software development...

"Low-fidelity (lo-fi) prototyping is characterised by a quick and easy translation of high-level design concepts into tangible and testable artefacts. Lo-fi is also known as low-tech, as the means required for such an implementation consist, most of the time, of a mixture of paper, cardboard, post-it notes, acetone sheets etc. A clear advantage of lo-fi prototyping is its extremely low cost and the fact that non-programmers can actively be part of the idea-crystallisation process." (Egger 2000)

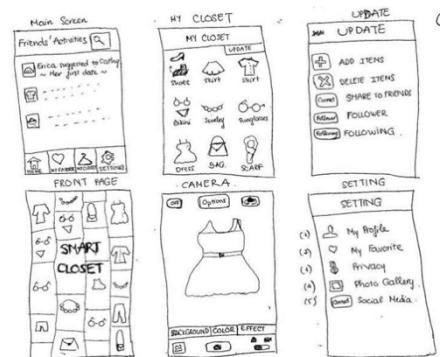


Figure 3: Low Fidelity Prototype Example

(Stolarski 2012)

Although low fidelity is the step at which hand-drawn sketches are created to give a representational view of the intended design to an audience that is not savvy in terms of technical skills, such as design in software such as Adobe Photoshop or Sketch. Low Fidelity designs also benefit those who are not strong in the software development area, it allows those who do not regularly use computing for productive purposes to give their opinion on the design before it is adapted further.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

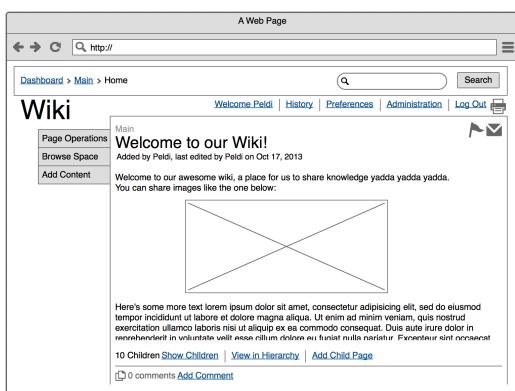


Figure 4: Mid Fidelity Prototype Example

(Carter 2015)

Mid fidelity design is the next step of the process and is usually catered with documents such as wireframes, these are higher quality designs but appear to be more of a skeleton than a fully-fledged design – this is a higher beneficial resource to UX or UI experts who understand the placement of specific controls and content based on the behaviours of a regular user of applications/websites/software, as explained by Mears, 2013.

The final development of the design process is to create a high fidelity design, this allows for a fully developed design to be brought to life via graphic design, this means that the design can be displayed as intended before production begins to create the final produced application/website/software which will hold functionality and be usable by the intended audience.

A definition of a high fidelity prototype as been given by Usability First...



Figure 5: High Fidelity Prototype Example

(Kapikul 2012)

"A prototype that is quite close to the final product, with lots of detail and functionality. From a user testing point of view, a high-fidelity prototype is close enough to a final product to be able to examine usability questions in detail and make strong conclusions about how behavior will relate to use of the final product." (Usability First 2010)

UX Documents

The UX documents that were created were generated specifically to outline the user journey, as expected to be followed by the developer, along with persona documents to cater for the specific user.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

The following documents were created to help understand the intended outcome of a visit from an expected user...

- 2 User-Type Personas
 - “User personas are fictitious users that are constructed using sample data that is collected from actual users.” (Paley 2015)
- Hierarchical Task Analysis
 - “In its most basic form, a hierarchical task analysis provides an understanding of the tasks users need to perform to achieve certain goals. You can break down these tasks into multiple levels of subtasks.” (Hornsby 2010)
- User Journey
 - “A user journey is a path a user may take to reach their goal when using a particular website. User journeys are used in designing websites to identify the different ways to enable the user to achieve their goal as quickly and easily as possible.” (Experience UX 2015)

Backend Design

The design of the dashboard and all backend accessible features of Hubbal (the final product), were originally designed through the low, mid and high fidelity design process, with reference to the moodboard’s discussed earlier to guide the design, the design process took a different direction due to the amount of time that was being forced towards the design creation process.

The change of methodology taken towards the backend design aspect was made to mitigate the risk of losing too much time in other aspects in the product, such as the production of the functionality and bringing Hubbal to life. As Hubbal is a virtual organisation, making reference to Grabowski and Roberts regarding the effective communication and effect on risk mitigation concretes the point further; “Risk mitigation in VO’s requires special attention to their peculiar characteristics.”

UX Documents

To cater towards the understanding, the user's journey and the tasks that they need to complete to achieve the applications desired tasks, in the backend of the Hubbal application, there needs to be special care made towards the multiple tasks the user needs to accomplish depending on their intended outcome they would like from the application.

User Personas

Due to the two separate views that have been generated within the Hubbal application, separate user personas needed to be created depending on which type of user is accessing a specific section of the application.

The expected users that will access the backend development section will purely be teachers or staff members of the organisation that have the application installed, as the project is catered towards Brune Park Community School, a persona has been created as an example teacher that will be using the application's backend.

Please see Appendix M for the Teacher User Persona aimed at the backend sector of the Hubbal application.

Hierarchical Task Analysis

There has been four hierarchical task analysis' designed to determine the planned route the user needs to take in order to complete the assigned tasks, these are all an overall view on a user performing tasks related to specific items, such as creating a user and a group, each task consists of the same steps, but with a different label of item that the task is being performed with.

Please see Appendix N for the group of Hierarchical Task Analysis' related to the Backend Design.

User Journey

Creating a user journey ensures that a theoretical situation that a user could land themselves in can be thought through to help re-iterate the importance of specific

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

features to the design, the user journey caters towards the design cycle of the project as an extra resource to help shape the way a design should look.

Please see Appendix O for the User Journey specific to the Backend Design of the Hubbal application.

Low Fidelity

To begin the low fidelity development of the design process, the fidelity approach was followed, beginning with the low fidelity stage; using the resources created within the UX documents phase of the design approach, the low fidelity version is hand-drawn and was created to showcase a basic design, after this, the design is then built upon through the fidelity levels to add detail.

Two designs were created in the low fidelity stage to give multiple options of design to take forward onto the mid fidelity stage.

Please see Appendix P for the Low Fidelity backend designs.

New Design Approach

As the timeframe for the project began to shorten to completion, the design segment for the backend section of the Hubbal application became very pushed for delivery on time, to keep within the time frame of the project expected to achieve the entire completion. After performing specific research into the usage OS software, beginning with the background basis of the benefits of OS software, explained by GBDirect in one complete document...

“Open Source's proponents often claim that it offers significant benefits when compared to typical commercial products. Commercial products typically favour visible features (giving marketing advantage) over harder-to measure qualities such as stability, security and similar less glamorous attributes.” (GBDirect 2004)

...after the specific research was made into the idea of open source software in terms of a design aspect, the decision was made within the project to implement an

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

open source design to the backend design, as the benefits of usability and approach towards the user-friendly experience would certainly be present; research was then made into available open source designs available that could be integrated into Ruby on Rails applications.

The choice of design that would be implemented into the application's backend zone was a Ruby gem specific to design labelled 'Admin LTE2', created by Almsaeed, 2015. The design is implemented and generated into the Ruby on Rails application using a Ruby Gem, as discussed previously in the 'Option Analysis' section of this report.

To see the Github repository containing the Admin LTE2 Ruby Gem, please see Appendix Q.

The development of the final design approach was built around the development process. Once production had begun on the backend section of the Hubbal application, the design was flexible due to the amount of options available within the OS Admin LTE2 design, simply by calling classes on elements created in the HTML view code would generate styles embedded into the design; taking away the pressure of adding stylings specific to the design, if it had been created by the project developer.

Please see Appendix R for the collection of high-fidelity designs of each page in the application, created using the Admin LTE2 open-source design Ruby gem.

Frontend Design

The frontend design was a much less technical approach because it did not need to be developed through any database or be embedded into the application's collection of model's which controls the backend data fed into the view. The frontend design followed the standard strategy of low-fidelity, mid-fidelity and high-fidelity.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Multiple (three) low fidelity concepts could be made, carrying over the two successfully chosen basic, low fidelity designs to be upgraded to mid-fidelity, finalising with taking one chosen design forward to become the high fidelity design.

UX Documents

To begin the process of designing the interface for the frontend zone of the application, the UX documents needed to be created to guide the design process to detail why specific pieces of the design have been created to the specifications of that the UX documents have outlined.

User Personas

Although the front end design is catered solely towards any user that accesses the internet and attempts to connect to the specific organisations website, the ability to determine a kind of user's persona that could access the frontend website allows for specification of the design to begin to take shape. The frontend of the website is purely content, but there are users who need such content.

Please see Appendix S for the frontend design example User Persona.

Hierarchical Task Analysis'

There has only been two hierarchical task analysis' created for the frontend design of the application, this is due to the type of user visiting the website should be there, typically to view pieces of content, whether related to the website itself, the school as a whole or even a specific course, the route to this will always be the same.

The second hierarchical task analysis that has been created is to outline the way a user would access and fill in a contact form to receive extra information from the organisation. In the example frontend designed for the sake of this project, the frontend design is for a school.

Please see Appendix T for the Hierarchical Task Analysis' for the frontend design of the application.

User Journey

The user journey has been created in order to detail an example scenario of a user accessing the frontend design, this means that the design can be determined and influenced in terms of creating a user experience devoted to the example journey drawn up in the early stages of the design process. The frontend design is flexible to any organisation which chooses to install Hubbal, so these types of user journey and UX documents are likely to be completely different based on the organisation that has Hubbal installed.

Please see Appendix U for the example User Journey used to help design the frontend example template.

Low Fidelity Designs

There were a total of three designs created within the low fidelity sector in order to give multiple options to choose from to design on a final frontend design, this will be the demonstration product represented to Brune Park Community School, meaning that the design needs to be appealing to the eye from a customer point of view and a user's point of view, if the demonstration website were to exist. The content in the design, even in final implementation will be very minimal, this will help keep within the timeframe specified in the Gantt chart produced in the early stages of the project life cycle.

Please see Appendix V for the group of Low Fidelity designs created for the frontend sector of the Hubbal application.

Mid Fidelity Designs

To demonstrate the importance of differentiating which designs were most important to take forward, the chosen designs were pushed to be transformed into mid-fidelity designs, which are wireframes rather than simple hand-drawn designs. These mid-fidelity designs were created using a tool known as Mockflow's WireframePro, a drag-and-drop wireframing tool with a brilliant graphical user interface, as described by Mockflow themselves...

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

"MockFlow WireframePro is a web-based tool to design and collaborate user interface blueprints for websites and apps. It helps to visualize the website's interface, navigation and structure in short time." ([Mockflow 2013](#))

...using this tool allows for the design to be brought to life quickly, whilst holding more detail than the low-fidelity designs, these are then analysed and moved onto the next phase, which is a high fidelity design phase.

Please see Appendix W for the Mid Fidelity designs created for the frontend sector of the Hubbal application.

High Fidelity Design

The final high fidelity design was pulled from the mid-fidelity range designs and was then created using an application known as Sketch, an application that is used to create vector imagery for use of template designing for multiple devices, as explained by [Sketch, 2014](#). The design was then created within Sublime Text using HTML, CSS and/or JavaScript if needed, implementation was then achieved through the template/layout feature embedded into Ruby on Rails.

Please see Appendix X for the High Fidelity design created for the frontend sector of the Hubbal application.

Issues Arising from Implementation and Test

Implementation and testing is very important to finalising the project and discovering whether or not the intended project aims and objectives have been met, the importance of implementation and testing within any project management is obvious and the implementation and testing must match the criteria or scenario that has been outlined within the project's early stages.

There are multiple tools available in today's age that offer new-comers to the web development industry plans to help along with their website implementation, [see Appendix Y for an example of this type of service, supplied by w3.](#)

Each section of the project has contributed a resource or material that will lead to the output implementation, these are generated pieces of content such as designs, research apparatus or documents of reference to build up the basic idea. This point is re-iterated by Chen and Kao, a pair of IEEE conference journal writers/attendees...

"Various stages of the project must generate tangible outputs in order to assess the implementation of the results and to determine whether there is a need to continue the project or the need for amendments." (Chen and Kao 2010)

Following from implementation will come the testing phase, used to ensure that all aspects of the aims and objectives have been met, setting these criteria gave a baseline for testing phases and attributes to be met against. As explained by Sean McManus in his Web Design book; "The goal of technical testing is not to create a perfect website (whatever that might mean). It is to give you confidence that there are no errors in how your site has been coded or configured, and that users don't have a bad experience because of technical issues, or when they make a mistake." (McManus 2012)

Implementation Phase

The implementation phase was conducted using the basic flow of the project lifecycle, whilst following the Agile methodology, the implementation phase

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

consisted of regular sprints at which product development took place, this began with the installation of the ‘ruby’ command upon the development computer. The primary computer that was used in order to develop the Hubbal application was an Apple iMac computer, after using research through online message boards/forums, the iMac appeared to be powerful enough to sustain web development and web design tasks.

Please see Appendix Z for an example message board topic detailing the benefits of using an iMac computer as a development-focused machine.



Figure 6: iMac Development Computer Specifications

As can be seen in Figure 6, the specifications of the iMac computer were more than enough to cater towards the production sector of the project, although leading on from this, developing within Ruby on Rails has specific technological dependencies that the computer does not come with by default.

Necessary Software Dependencies

The particular software dependencies that are important to the project must be installed on the development machine before any production can begin, through using search results to discover how Ruby on Rails can be installed on a Mac computer returned the following results, as seen in Figure 7 below.

A screenshot of a Google search results page. The search query "Installing ruby on rails" is entered in the search bar. Below the search bar, there are filters for "All", "Videos", "News", "Shopping", "Images", "More", and "Search tools". The "All" filter is selected. Below the filters, it says "About 487,000 results (0.61 seconds)". The first result is titled "Rails Installation on Windows" and includes a list of 8 steps. To the right of the title is a screenshot of the RailsInstaller website. Below the list of steps is a link to "Ruby on Rails Installation - Tutorialspoint" with the URL "www.tutorialspoint.com/ruby-on-rails/rails-installation.htm". At the bottom of the result card is a link "About this result • Feedback".

Rails Installation on Windows

1. Step 1: Check Ruby Version. First, check if you already have Ruby installed. ...
2. Step 2: Install Ruby. ...
3. Step 3: Install Rails. ...
4. Step 4: Check Rails Version. ...
5. Step 1: Install Prerequisite Dependencies. ...
6. Step 2: Install rbenv. ...
7. Step 3: Install Ruby. ...
8. Step 4: Install Rails.

More items...

Ruby on Rails Installation - Tutorialspoint
www.tutorialspoint.com/ruby-on-rails/rails-installation.htm

Figure 7: Ruby on Rails Installation Search Results

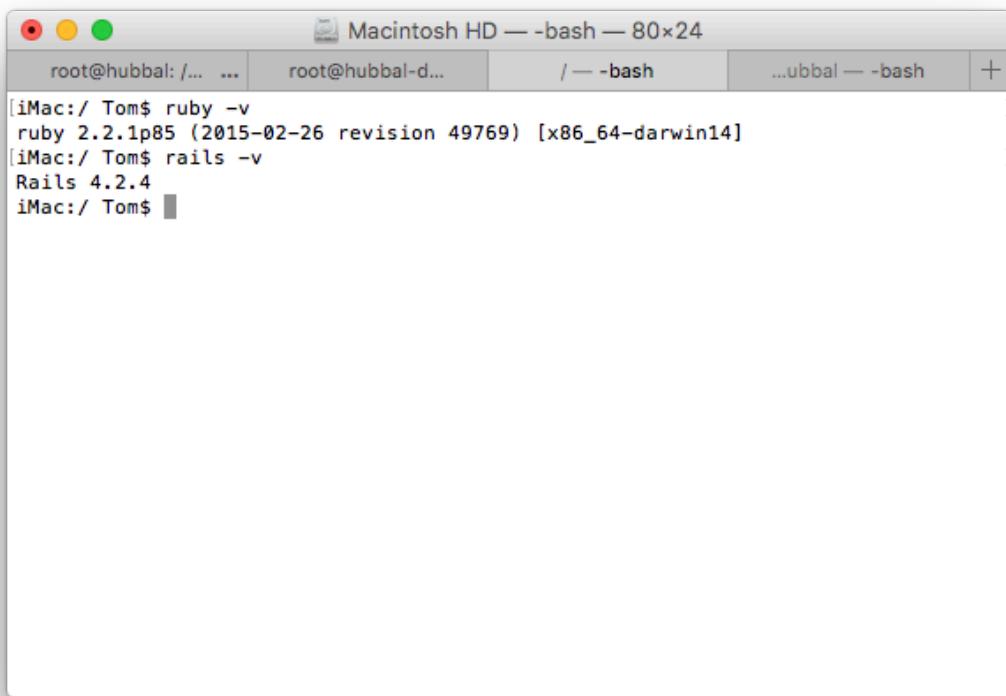
(Google 2016)

As can be seen, 'RailsInstaller' was the first result that has been returned by the Google search results and was visited, the website for RailsInstaller was returned and was described by RailsInstaller as to what it is; "RailsInstaller is the quickest way to go from zero to developing Ruby on Rails applications. Whether you're on Windows or Mac, RailsInstaller has you covered." (RailsInstaller 2013)

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Once RailsInstaller was completed with its installation, the usage of the ‘ruby’ command and the ‘rails’ command was available for use on the development iMac computer through the terminal, although these commands have sub-commands available to perform specific tasks inside the ‘ruby’ or ‘rails’ environment. A cheat sheet was found in order to help understand what commands are available and what each command does.

Please see Appendix AA for the Rails & Ruby command cheat sheet.



The screenshot shows a Mac OS X terminal window titled "Macintosh HD — bash — 80x24". The window has four tabs at the top: "root@hubbal: /... ...", "root@hubbal-d... / — bash", "...ubbal — bash", and "+". The main pane displays the following terminal session:

```
[iMac:/ Tom$ ruby -v
ruby 2.2.1p85 (2015-02-26 revision 49769) [x86_64-darwin14]
[iMac:/ Tom$ rails -v
Rails 4.2.4
iMac:/ Tom$ ]
```

Figure 8: iMac Command Line with Ruby and Rails Version

As can be seen above, the ‘ruby’ and ‘rails’ commands were recognised by the iMac development computer and were specified as ruby version ‘2.2.1p85’ and rails version ‘4.2.4’ – these are the highest versions of ruby and rails available and are perfect for the intended purpose of this project development cycle.

Software Creation

Considering the intended usage of the Hubbal application is to run locally, on a network in development mode as to allow any changes to be made to the application whenever the client feels they need to, the development cycle was performed through the iMac development computer, which will mimic the ability of a server that would be installed on the local network of the organisation running the application.

To also serve as a hosting environment to ensure that the application's core files is accessible across multiple devices, in the event that anything is to happen to the development computer, Dropbox was used to ensure that all files were instantaneously synced to a 'cloud' server space to secure files during development. For those who do not know, Dropbox is a service that allows you to share data between computers and other devices by storing files on its servers, as explained by Cohen, Cohen and Spangenberg, 2011.

Please see Appendix BB for the output of the 'rails new' command.

As seen in Appendix BB, after locating the 'Dropbox' folder in the terminal, to firstly create the application, the usage of the 'rails new' command needed to be entered to allow rails to generate the files to run the Hubbal application, this command created a folder containing all of the files from start to finish under the folder named after the name of the application entered.

Once the files for the application have been created, the database must be generated by creating each individual table, the usage of the 'rails generate scaffold [table]' was used in order to generate the database table to store the information for each set of results, along with a model, view and controller for each individual table created in the database. The database is a SQLite database and comes stocked with Ruby on Rails as standard, described by SQLite themselves...

"SQLite is an in-process library that implements a self-contained, serverless,zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

purpose, commercial or private. SQLite is the most widely deployed database in the world with more applications than we can count, including several high-profile projects." ([SQLite 2007](#))

Please see Appendix CC for the results of the 'rails generate scaffold' command for a User, the same process was followed for each table containing data.

As can be seen in Appendix CC, each file has been generated to allow the application to run for a user, this includes files that can specify the CSS, HTML and JS that is attached to the specific collection of views, each model created in a scaffold contains the views labelled 'index.html.erb', 'show.html.erb', 'edit.html.erb', '_form.html.erb' and 'new.html.erb' – these files all contain content needed to control creation, deletion and amendments of a specific user, along with listing all users.

With the created files in the scaffold, the syntax of Ruby when embedded in HTML is setup as an example to help understand how simply displaying a record or set of records can be.

To see these generated files, please see the attached evidence in Appendix DD.

There were files generated for each table, please see below the table outlining the created tables with their attributes/columns...

Table 2: Database Users Table

Table Name: Users	
Name	Data Type
firstname	String
lastname	String
username	String
password_digest	String
email	String

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

phone	String
role	Integer
status	Integer
room	String

Table 3: Database Contents Table

Table Name: Contents	
Name	Data Type
course_id	Integer
html	Text

Table 4: Database Enrolments Table

Table Name: Enrolments	
Name	Data Type
user_id	Integer
course_id	Integer

Table 5: Database Groups Table

Table Name: Groups	
Name	Data Type
name	String
leader_id	Integer

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Table 6: Database Login Activities Table

Table Name: Login_Activities	
Name	Data Type
time	Date/Time
ip	String
user_id	Integer

Table 7: Database Messages Table

Table Name: Messages	
Name	Data Type
recipient_id	Integer
sender_id	Integer
content	Text
subject	Text

Table 8: Database Timetables Table

Table Name: Timetables	
Name	Data Type
user_id	Integer

Table 9: Database Timetable Entries Table

Table Name: Timetable_Entries	
Name	Data Type
timetable_id	Integer
lesson_id	Integer

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

start	Date/Time
finish	Date/Time
day	String
room	String

This then allowed for each table to be created along with the files to accompany the control of the table, this allows users to add rows and remove them if they please, creating the data control aspect of the application to start with. Following on from this, the design aspects need to be created. Following on from this creation, the design gem labelled 'Admin LTE2' was then integrated into the application via the Gemfile, along with specific features such as Paperclip for avatar support, as well as features such as Bootstrap and jQuery.

Please see Appendix EE for the Gemfile located in the application directory.

The command known as 'bundle install' was then ran in the applications root folder via the Terminal to install all of the new dependences added to the Gemfile, although particular gem's were not compatible and resulted in a Ruby update needing to be performed within the iMac computer being used for development.

Ruby on Rails supports multiple layouts and templates through the use of a layouts folder, located in the 'views' folder of the application along with all the other HTML files generated in the rails generate scaffold sector of production. The structure of specifying layout to render within a controller in Ruby on Rails has been specified within the Ruby on Rails documentation, along with the branches followed should no layout be specified for a particular controller/view...

"To find the current layout, Rails first looks for a file in app/views/layouts with the same base name as the controller. For example, rendering actions from the PhotosController class will use app/views/layouts/photos.html.erb (or app/views/layouts/photos.builder) . If there is no such controller-specific layout, Rails will use app/views/layouts/application.html.erb or app/views/layouts/application.b

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

uilder. If there is no .erb layout, Rails will use a .builder layout if one exists. Rails also provides several ways to more precisely assign specific layouts to individual controllers and actions.” ([Rails Guides 2010](#))

Setting these layouts that have been created by the AdminLTE2 gem and how they were put to use can be found in the Design section of this report.

Each table created needed specific functions created to access pieces of data to be used within each view, these were done in steps throughout the entire process by following the Agile methodology planned in the early stages of the project definition document. By following the same method of creating specific functions in the model for each table, the process of understanding which functions needed to be created become much clearer as the project evolved.

Please see Appendix FF for the Application model for each table.

As discussed earlier, each database table created is accompanied with a model, view and controller, the controller files were left unchanged throughout the entire project, besides specific layout rendering which is a simple code labelled ‘render :layout => “modal”’ to specify that the page will be laid out to be used in a modal, built into the Bootstrap dependency gem.

Layouts are rendered within the controller’s, for each table, a controller is created to control the views of each model, controllers can pass over variables or parameters and also control the interaction with the database, such as creating, editing and/or destroying records within the database, depending what parameter has been passed over to the controller. The controllers were left untouched throughout the product development, besides specifying new views to be pointed at along with rendering new layouts.

Please see Appendix GG for an example of the User controller generated by Ruby on Rails in the ‘rails generate scaffold’ command.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

A custom 'modal.js' file was created to generate a function using jQuery that uses the data attached to a link and ensures that it is loaded into a modal, rather than loaded as an ordinary page. Please see Appendix EE for the JS file that handles the ability of loading a page inside a Bootstrap modal component using AJAX.

Please see Appendix HH for the modal.js source code.

Frontend Integration

The frontend integration was an additional feature discussed in the Design section of this report, after the design phase was completed and the backend implementation was complete, the front end integration needed to be created, this is a feature which was intended to give the website a frontend for visitors of the organisation's website to view – this would be if the organisation wanted the application to be external, rather than internal to their local network.

Implementation

The frontend integration meant creating a layout/template file that could wrap around the interchangeable content that was specified by the view files in the '/frontend' folder, the layout was built using HTML and CSS only to cater for progressive enhancement purposes, with a multitude of users viewing websites daily, specific support needs to be tunnelled to cater for all users.

Pettit, a Team Tree House blogger, gives an excellent definition to what progressive enhancement is...

"Progressive enhancement is a layered approach for building websites using HTML for content, CSS for presentation, and JavaScript for interactivity. If for some reason JavaScript breaks, the site should still work and look good. If the CSS doesn't load correctly, the HTML content should still be there with meaningful hyperlinks. And even if the HTML is malformed, browsers are smart enough to continue rendering what they can." (Pettit 2014)

To view the source code for the implemented front-end design, please see
Appendix II.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

As can be seen in Appendix II, the use of ruby has been entered into the navigation, as the frontend website is a demonstration of a college website, known as Silverleaf College, once a course has been created within the backend of the application, the course is then added to the navigation after checking the database and displaying the results returned from the 'courses' table.

Route Manipulation

Within a Ruby on Rails application, there is a route file, this file is used to determine which file is the root file of the entire application

and can point users to different locations and areas, usually used to manipulate url's to make a user believe they are on one page, when really they are on another.

```
root :to => 'frontend#index'  
get '/about' => 'frontend#about'  
get '/history' => 'frontend#history'  
get '/location' => 'frontend#location'  
get '/support' => 'frontend#support'
```

Figure 9: Routes.rb File Example

This route file has setup the frontend application to make users believe that held within these /about, etc URL's are the true pages, although they are really held within the /frontend folder inside the views section of the application.

Issues During Implementation

```
20160403115944_remove_timetable_id_from_groups.rb — portfolio  
1 class RemoveTimetableIdFromGroups < ActiveRecord::Migration  
2   def change  
3     remove_column :groups, :timetable_id, :integer  
4   end  
5 end
```

Figure 10: Removing Timetable ID from Groups

As the product slowly came to life, a feature was changed, timetable's were changed from being group-specific to becoming user-specific, to help organisation, this meant using 'migrations'. Migrations

are used to migrate changes from the command line to the database if any changes

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

need to be made. By using the command 'rails generate migration [migration name]' a file is generated to specify what changes need to be made to the database. A migration was made to drop the column known as 'timetable_id' from the 'Groups' table, this can be seen in Figure 9 to the left.

Next, another migration file was generated labelled 'AddTimetableIDToUsers', this generated the migration perfectly, although prior to this working well, there was an incident at which a limit was accidentally set to number of '100000', this resulted in the application being unable to load and instantly failed.



```
1 class AddTimetableIDToUsers < ActiveRecord::Migration
2   def change
3     add_column :users, :timetable_id, :integer
4   end
5 end
```

Figure 11: Adding TimetableID to Users

This issue was repaired by backing up the iMac development computer to a week prior, although this caused issues with loss of data and development, but being ahead of schedule assured the developer that there were no issues in terms of losing time. By using the correct documentation to track progress, the features were re-added once again.

The final deployment of the demonstration product is located at <http://139.59.165.197:3000> - a private server used to emulate a local network of an organisation known as Silverleaf College, this is the demonstration product used for testing and demonstration to introduce to Brune Park Community School.

Testing Phase

Testing is a fundamental sector of any project and without testing, there would no measurement of success to give a result on whether the finished product has achieved its original intended purpose. As the project has been running through the Agile methodology, it's important to understand how necessary testing is within the Agile methodology life cycle, as explained by Crispin...

"One of my favorite quotes from Elisabeth Hendrickson is this: "Testing is not a phase!" We have to stop thinking of "testing" as some separate activity,

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

removed from the rest of software development. It takes a village to produce a high-quality software product, and we can't do it from isolated functional silos. The idea of driving development with tests has been popularized by the agile development movement." (Crispin 2009)

To fully analyse the product and discover whether it has ticked the correct boxes of completion or not, the following techniques have been used to determine the level of success the product has reached...

- Device Testing
- Criteria for Success Results

Combined together, the analysis of the product completion can be illustrated to a very high degree, even by those who have not read or researched into the project development stage.

Device Testing

Using devices to test with can be very beneficial to the workflow within a project, although there are arguments for and against testing with emulated device environments or using real devices that your users will be accessing your application with. As explained by Kelly below, real devices are much more beneficial to the real users who will be using the application...

"Eran Kinsbruner, mobile technical evangelist at Perfecto Mobile, a mobile app testing platform provider, stresses that testing mobile apps on emulators only is a huge risk because your users are running apps on real devices, not emulators. There are things that you just can't find on emulators and simulators, and that's one of the biggest reasons that people use them in conjunction with real devices." (Kelly 2016)

The choice of testing that was used after deploying the Hubbal application was the usage of real devices, as this is a much more beneficial test to give real results that can be witness and fixed as soon as they appear.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

The devices that were used to test the Hubbal application were...

1. Apple iMac Computer (2012)
2. Apple MacBook (2015)
3. Windows Acer Laptop
4. iPad 2
5. iPad Air 2
6. iPhone 5S
7. iPhone 6
8. iPhone 6 Plus

The testing on each of these devices returned without any issues in the application what-so-ever, this is mainly due to the fact that testing was periodically implemented upon completion of each feature/view with the Google Chrome browser, meaning any issues could be cleared up in development before testing had begun.

Please see Appendix JJ for the evidence of testing on the above eight devices.

Criteria for Success Results

Criteria for success is helpful when determining the level of success, the project has reached, although pictures/photos are a great asset to use to prove points in authenticity of completion, facts and figures can be just as important in proving how far the project has come through completion.

The importance of a criteria for success is explained by TutorialsPoint, a website devoted to project management description; "A project success criterion begins with the initiatives taken by the project manager to the project in question. This will increase the chances of the project becoming successful as well as meeting customer's expectations." (TutorialsPoint 2010)

Criteria for success was created in early development of the research segment of the project to paint a picture of what the final outcome should be once the project is at

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

level of completion, included in the project definition. The final criteria for success document can be seen below...

Table 10: Criteria for Success Results

Criteria	Met?	Rating of Completion (/5)	Comments
Responsive, able to change dependent on user's device.	Yes	5	N/A.
Progressively Enhanced, works on all supported browsers.	No	2	The system relies on JavaScript in specific places.
Informal content, laid back approach to user direction.	Yes	4	Users are approached from a friendlier manner than formal, this is to ensure that users feel at home, more than at work.
Modal forms, allows user to appear on the same page.	Yes	5	N/A.
Login is fully secure with necessary encryption.	Yes	5	Ruby on Rails has bcrypt encryption built into database entries labelled "password_digest".
Colour scheme is neutral and calm to the user.	Yes	4	N/A.
Uses Ruby on Rails technologies.	Yes	5	N/A.
Creates usernames based on first name and last name unique to the user.	Yes	5	The first name and last name is reconstructed before saving the username to the database to create a mixture of letters and

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

			numbers for the username.
Timetable Booking	Yes	5	N/A.

Evaluation & Conclusions

Concluding the project will consist of judging each sector of the project, along with the chosen technique/methodology of choice to produce the specific area of the project being discussed, featuring the effect it had on the project, definition as to why this option was chosen and whether or not the desired outcome was met by using said option to achieve the sector of choice.

The conclusion/final analysis in any project is of higher importance than any other moment previously discussed within the report, for those who do not feel the project is of their particular interest or taste, the conclusion gives the overview and outcomes of the project in a much smaller format. Although conclusions are primarily short and consist of a singular paragraph, as the size of this project has developed through production, the conclusion has also grown in unison with every other aspect previously mentioned.

Overview

This report has discussed, analysed and followed the production of the application known as Hubbal, a web-based application built with the sole purpose of organising a company's staff/faculty on a weekly basis, intended to run privately on a local network. By using a school known as Brune Park School to introduce the project intentions as an outline to follow to shape the final project, the demonstration product was eventually brought to life to use as an example to those interested in the application and can be found at <http://139.59.165.197:3000/> as stated in the implementation documentation.

Aim

The original aim of the project set out in the earliest stages during the projects birth, the intended aim was as follows...

"To build personal programming and server development skills to manufacture an Attendance Tracking software that is capable of use in an

educational organisation environment for tracking and timetabling staff's personal daily schedules."

To re-iterate the aim, the application is focused primarily towards the educational institution's that would have the ability to use such an application along with furthering the developers programming and server development skills. It's safe to say that this aim was met, with programming skills being enhanced through the evidence provided in the implementation stage, along with the source code that is also available to those reading.

The scheduling and timetabling feature to provide with the ability to organise their daily schedules was certainly achieved, as can be seen in the demonstration product, the available options a User has is to control their own Timetable, with adding their own events based on the available courses, with the added functionality of viewing a Users profile to view their timetable/schedule for the current day.

Please see Appendix KK for the pages in the application demonstrating the Timetabling functionality.

Objectives

The objectives that were outlined in the early stages of the project were created to be a measurable achievement to determine whether or not the project had truly achieved its intended purpose. The objectives that were listed in the project specification were as follows...

- Secured accessibility, using login and high level encrypted passwords.
- Accessible Web Interface, scalable to be accessed on any device for portable usage when connected to network controlling application.
- Sustainable on a system with a maximum of 200 user accounts and regular connections.
- Primary support for top five most used web browsers, judging by global web statistics.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Each of these points were created to allow the project to be judged on its level of completion, depending on the amount of objectives that have been met, the determination of the completion of the project can be measured.

Security

This section in particular pays evidence to the following objective...

“Secured accessibility, using login and high level encrypted passwords.”

The need for security in systems containing user data is very high and it must be ensured that the system is unable to be attacked or easily accessed by those without a user account – to enable this feature, a sessions controller was created, which is used to primarily login, also the use of the application_controller.erb file ensured that all pages needed a session containing the user’s login information to be set within their browser before they could access the page.

```
def current_user
  @current_user ||= User.find(session[:user_id]) if session[:user_id]
end

def require_user
  redirect_to '/login' unless current_user
end
```

Figure 12: Evidence of Page Security

As it can be seen in Figure 12 above, the methods that have been created in the application_controller file ensure that unless a current_user is set or can be referenced to, depending if a session is set, the page cannot be accessed by any random user, unless an exception is made in that particular pages controller.

The usage of the ‘password_digest’ ability in Ruby on Rails was also used, by creating users with this field in the database, Ruby on Rails creates two password fields, one labelled password as a string and another known as password_confirmation, also stored as a string and also uses the bcrypt algorithm to achieve full irreversible string generation.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Please see Appendix LL for an article giving further discussion and explanation on the ‘password_digest’ option in Ruby on Rails.

Responsive Design

The objective of responsive design was created to justify the usage of the application across multiple devices, this would allow users of the institution with the Hubbal application installed to be able to use the application on any sized device within the school or business.

The objective that responsive design capabilities are made to fulfil is the following...

“Accessible Web Interface, scalable to be accessed on any device for portable usage when connected to network controlling application.”

As described in the design sector and the testing phase, responsive design has been successfully met through the use of the Admin LTE2 gem that was added to the application, along with the testing taking place on over 5 devices with all different sizes and returning a positive feedback in terms of change in design. An example of the benefit of this would be if an employee were to need a portable device, such as a tablet or mobile phone to access the application if they were perhaps traveling around the campus of the organisation or busy elsewhere.

Sustainability & Support

Sustainability is to be measured through the use of the following objective...

“Sustainable on a system with a maximum of 200 user accounts and regular connections.”

Although the system was tested with a limiting amount of users, the overall system was expected to be designed to cater for a multitude of users, with the maximum being two hundred and above, although anymore than two hundred would need to be justified with the correct technology to do so.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

With ongoing connections being made throughout the usage of the testing phase with multiple devices connected to a singular server, the testing of sustainability has been met within the regular connections expectations.

The support objective of five browsers was accomplished through the testing phase whilst using separate devices to test the system within.

Recommendations

The project was completed to a standard that exceed expectations of the originally documented specification and the necessary amendments to the documentation was made through each development stage to keep up with the rapidly growing interface and features.

Methodology

The chosen methodology for this project was Agile, the only issue with Agile, which was slowly discovered throughout the projects development was that, Agile is a methodology built primarily for team based projects and with the Hubbal application development being developed by one team member, the sprints and work load began to pile up as the project developed.

Ultimately, if the project was to be repeated again, Agile methodology would not be the way to go, the choice in future when taking on a project such as this, would be Waterfall or a more conventional, traditional type of methodology.

Technology

The use of Ruby on Rails as the primary technology to use as the catalyst to bring Hubbal to life was a challenging attempt, developing an application in Ruby on Rails without prior experience in it's technologies from an academical creating a learning curve for the developer. Whilst research was taking place, it was apparent that Ruby on Rails was perfect for the use of local development – although vast knowledge in the format Ruby on Rails produces content and file structure is needed prior to providing maintenance or development to the existing product.

Previous knowledge in PHP ("PHP is an HTML-embedded scripting language. Much of its syntax is borrowed from C, Java and Perl with a couple of unique PHP-specific features thrown in. The goal of the language is to allow web developers to write dynamically generated pages quickly." ([PHP.NET 2001](#))) produced from a academical background would have been the most beneficial route to go down, and most

certainly would have shortened the project timeframe by a substantial amount, although this would not have taught the developer a new technology.

Design

Design within this project was an issue that caused time lapses and was the only segment which caused risk mitigation to be called upon to solve the issue, if repeated again, the design segment of the project would have had more time devoted to it, as due to the lack of judgement by the project developer during the time management planning phase, not enough time was devoted to design and thus resulted in the use of open-source software, causing the project to feel less unique upon completion.

References

1. ALDERSON, C., 2015.
2. ALMSAEED, A., 2015. Features of AdminLTE 2.1. In: *Almsaeed Studies*. 15 June 2015 [viewed 23 April 2016]. Available from: <https://almsaeedstudio.com/blog/features-of-adminlte-2.1>
3. ANTHONASIOU, T., DEBAS, H. and DARZI, A., 2010. *Key Topics in Surgical Research and Methodology*. London, New York: Springer Heidelberg Dordrecht
4. BIEN, H., 2013. 4 Reasons Why You Should Learn Ruby As Your First Programming Language. In: *Skilled Up - Cloud Computing Blog*. 11 May 2013 [viewed 15 April 2016]. Available from: <http://www.skilledup.com/articles/4-reasons-learn-ruby-first-programming-language>
5. CARTER, M., 2015. *Mid Fidelity Prototype Example* [digital image] [viewed 21 April 2016]. Available from: <http://morgancarter.com.au/assets/images/blog/prototypes/medium-fidelity.png>
6. CHARITIES EVALUATION SERVICES., n.d. *Aims, objectives, outcomes and impact* [viewed 17 April 2016]. Available from: <http://www.ces-vol.org.uk/about-performance-improvement/about-monitoring-evaluation/planning-for-monitoring-evaluation/aims.html>
7. CHEN, Y. L. and KAO, H. C., 2010. Notice of Retraction. *The effects of strategic implementation of project management and performance*. 1(1), 1-2.
8. CLARK, W., 1923. *The Gantt Chart – A Working Tool of Management*. 2nd Edition. New York: The Ronald Press Company
9. COHEN, E. M., D. R. COHEN and L. L. SPANGENBERG , 2011. *The iPad 2 Project Book*. Location Unknown: Pearson Education
10. CRISPIN, L., 2009. *Coding and Testing: Testers and Programmers Working Together* [viewed 26 April 2016]. Available from: <http://www.methodsandtools.com/archive/archive.php?id=88>

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

11. CUTERANK., 2010. *What is HTML? A Simple Definition* [viewed 17 March 2016]. Available from: <http://cuterank.net/html-definition/>
12. EGGER, N, F., 2000. *Lo-Fi vs. Hi-Fi Prototyping: how real does the real thing have to be?* [viewed 21 April 2016]. Available from: <http://www.telono.com/en/articles/lo-fi-vs-hi-fi-prototyping-how-real-does-the-real-thing-have-to-be/>
13. EXCEL EASY., n.d. *Figure 1: Gantt Chart Example* [digital image] [viewed 19 April 2016]. Available from: <http://www.excel-easy.com/examples/images/gantt-chart/gantt-chart-second-result.png>
14. EXPERIENCE UX., 2015. *What are user journeys?* [viewed 21 April 2016]. Available from: <http://www.experienceux.co.uk/faqs/what-are-user-journeys/>
15. FORSYTH, P., 2010. *Successful Time Management*. Revised 2nd Edition. London, Philadelphia and New Delhi: KoganPage.
16. FREE MANAGEMENT EBOOKS., 2014. *SMART - Specific Measurable Achievable Realistic Timely* [viewed 17 April 2016]. Available from: <http://www.free-management-ebooks.com/faqps/goal-04.htm>
17. GALLUD, A, J., TESORIERO, R. and PENICHET, R, M, P., 2011. *Distributed User Interfaces: Designing Interfaces for the Distributed Ecosystem*. London, Dordrect, Heidelberg and New York: Springer-Verlag
18. GBDIRECT., 2004. *Benefits of Using Open Source Software* [viewed 23 April 2016]. Available from: <http://open-source.gbdirect.co.uk/migration/benefit.html>
19. GOOGLE., 2016. *Figure 7: Ruby on Rails Installation Search Results* [digital image] [viewed 26 April 2016]. Available from: <https://www.google.co.uk/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=Installing+ruby+on+rails>
20. GRABOWSKI, M. and ROBERTS, H, K., 2006. Risk Mitigation in Virtual Organisations. *Journal of Computer-Mediated Communication*, 3(4), 3.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

21. HANSSON, H. D., 2009. *Getting Started with Rails – Ruby on Rails Guides* [viewed 15 April 2016]. Available from: http://guides.rubyonrails.org/getting_started.html
22. HORNSBY, P., 2010. Hierarchical Task Analysis. In: *Innovating UX Practise Article*. 8 February 2010 [viewed 21 April 2016]. Available from: <http://www.uxmatters.com/mt/archives/2010/02/hierarchical-task-analysis.php>
23. JEFFREY, 2014. *I suggest you...* [viewed 19 April 2016]. Available from: <http://help.mightytext.net/forums/123495-general/suggestions/5509836-do-not-sacrifice-speed-for-features>
24. KAPIKUL, M., 2012. *High Prototype Example* [digital image] [viewed 21 April 2016]. Available from: <http://miodragkapikul.com/Work/choreo.html>
25. KEHOE, D., 2013. *What is Ruby on Rails? · RailsApps* [viewed 15 April 2016]. Available from: <http://railsapps.github.io/what-is-ruby-rails.html>
26. KELLY, W., 2016. Mobile App Testing: When to use real devices versus emulators. 12 February 2016 [viewed 26 April 2016]. Available from: <http://techbeacon.com/mobile-app-testing-benefits-when-use-real-devices-vs-emulators>
27. LAYTON, M., 2013. The Function of the Scrum and Sprint within an Agile Project. In: *For Dummies*. 5 December 2013 [viewed 19 April 2016]. Available from: <http://www.dummies.com/how-to/content/the-function-of-the-scrum-and-sprint-within-an-agile.html>
28. MARTINELLI, J. R., MILOSEVIC, Z. D., 2016. *Project Management Toolbox: Tools and Techniques for the Practicing Project Manager*. 2nd Edition. Hoboken, New Jersey: John Wiley & Sons, Inc.
29. MCMANUS, S., 2012. *Web Design in Easy Steps*. 5th ed. United Kingdom: In Easy Steps Limited.
30. MEARS, C., 2013. Wireframes – A Beginner’s Guide. In: *The UX Review*. 25 April

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

- 2013 [viewed 21 April 2016]. Available from:
<http://theuxreview.co.uk/wireframes-beginners-guide/>
31. MERRIAM-WEBSTER., 2007. *Information Technology / Definition of Information technology by Merriam-Webster* [viewed 17 April 2016]. Available from:
<http://www.merriam-webster.com/dictionary/information%20technology>
32. MICROSOFT., 2011. *SQL Server 2005 Express Edition Overview* [viewed 17 April 2016]. Available from: [https://technet.microsoft.com/en-us/library/ms345154\(v=sql.90\).aspx](https://technet.microsoft.com/en-us/library/ms345154(v=sql.90).aspx)
33. MOCKFLOW., 2013. *MockFlow WireframePro - Design user interface concepts for websites* [viewed 23 April 2016]. Available from:
<https://mockflow.com/apps/wireframepro/>
34. MOVEABLE ONLINE., 2013. 10 Advantages of Using a CMS to Run Your Site. In: *Moveable Online*. 29 October 2013 [viewed 19 April 2016]. Available from: <http://moveableonline.com/blog/2013/10/29/7-advantages-using-cms-run-site/>
35. MOZILLA DEVELOPERS NETWORK., 2015. CSS3 – CSS | MDN. In: *Mozilla Development Network*. 26 April 2015 [viewed 17 March 2016]. Available from: <https://developer.mozilla.org/en/docs/Web/CSS/CSS3>
36. NET SOLUTIONS., 2012. *Distributed Agile Development* [viewed 19 April 2016]. Available from: <http://www.netsolutionsindia.com/distributed-agile>
37. NORMAN, D. and NEILSEN, J., 2013. *The Definition of User Experience* [viewed 21 April 2016]. Available from: <https://www.nngroup.com/articles/definition-user-experience/>
38. PACHECO, M., 2014. Understanding Design Fidelity for Creating a Great Product Experience. In: *Cantina*. 27 February 2014 [viewed 19 April 2016]. Available from: <https://cantina.co/understanding-design-fidelity-for-creating-a-great-product-experience/>
39. PALEY, A., 2015. The Advantages of Building a User Persona. In: *Blue Fountain Media Blog*. 4 November 2015 [viewed 21 April 2016]. Available from:

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

<http://www.bluefountainmedia.com/blog/the-advantages-of-building-a-user-persona/>

40. PCMAG., 2013. Javascript Definition from PC Magazine Encyclopaedia. In: *PC Mag Encyclopaedia*. 9 May 2013 [viewed 17 March 2016]. Available from: <http://www.pc当地.com/encyclopedia/term/45585/javascript>
41. PETTIT, N., 2014. Progressive Enhancement: Past Present and Future. In: *Team Treehouse Blog*. 17 November 2014 [viewed 26 April 2016]. Available from: <http://blog.teamtreehouse.com/progressive-enhancement-past-present-future>
42. PLONE., n.d. *What is a CMS? — Plone CMS: Open Source Content Management* [viewed 19 April 2016]. Available from: <https://plone.org/documentation/faq/what-is-a-cms>
43. RAILSGUIDES., 2011. *The Asset Pipeline – Ruby on Rails Guides* [viewed 18 April 2016]. Available from: http://guides.rubyonrails.org/asset_pipeline.html
44. RAILSGUIDES., 2010. *Layouts and Rendering in Rails* [viewed 26 April 2016]. Available from: http://guides.rubyonrails.org/layouts_and_rendering.html#using-render
45. RAILSINSTALLER., 2013. *RailsInstaller* [viewed 26 April 2016]. Available from: <http://railsinstaller.org/en>
46. RANGAHAU., 2008. *Why do you need a methodology?* [viewed 19 April 2016]. Available from: <http://www.rangahau.co.nz/methodology/55/>
47. ROUSE, M., 2014. What is HTML5? – A Definition from WhatIs.com [viewed 17 March 2016]. Available from: <http://searchsoa.techtarget.com/definition/HTML5>
48. RUBYGEMS., 2011. *Guides – RubyGem Guides* [viewed 17 April 2016]. Available from: <http://guides.rubygems.org/>
49. SKETCH., 2014. *Sketch – Professional Digital Design for Mac* [viewed 23 April 2016]. Available from: <https://www.sketchapp.com/>

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

50. SOLENT UNIVERSITY., n.d. *Writing Dissertations: Aims and Objectives* [viewed 15 April 2016]. Available from:
<http://mycourse.solent.ac.uk/mod/book/view.php?id=116233&chapterid=152>
94
51. SQLITE., 2007. *About SQLite* [viewed 26 April 2016]. Available from:
http://web.archive.org/web/*/https://www.sqlite.org/about.html
52. STOLARSKI, L., 2012. *Low Fidelity Prototype Example* [digital image] [viewed 21 April 2016]. Available from: http://resources.infosecinstitute.com/wp-content/uploads/112712_2207_LowFidelity1.jpg
53. TECH TERMS., 2008. *Open Source Definition* [viewed 23 April 2016]. Available from: <http://techterms.com/definition/opensource>
54. THE POSTGRESQL GLOBAL DEVELOPMENT GROUP., 2004. *PostgreSQL: About* [viewed 17 April 2016]. Available from:
<http://www.postgresql.org/about/>
55. TOP ACHIEVEMENT., 1999. *Creating S.M.A.R.T. Goals – Top Achievement* [viewed 17 April 2016]. Available from:
<http://topachievement.com/smart.html>
56. TUTORIALSPOINT., 2010. *Project Success Criteria* [viewed 26 April 2016]. Available from:
http://www.tutorialspoint.com/management_concepts/project_success_criteria.htm
57. USABILITY FIRST., 2010. *Usability First – Usability Glossary – high fidelity prototype / Usability First* [viewed 21 April 2016]. Available from:
<http://www.usabilityfirst.com/glossary/high-fidelity-prototype/>
58. VILLANUEVA, J., 2011. *Optional Analysis Process* [viewed 17 April 2016]. Available from: <http://www.slideshare.net/BPArchitect/option-analysis-process>

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

59. W3SCHOOLS., 2000. *CSS Introduction* [viewed 17 March 2016]. Available from:
http://www.w3schools.com/css/css_intro.asp
60. WALCZAK, M., 2015. How to define the SMART goal technique. In: *Search Party Blog*. 30 June 2015 [viewed 17 April 2016]. Available from:
<http://www.thesearchparty.com/blog/definition-smart-goals/>
61. WATERS, K., 2007. 10 Good Reasons To Do Agile Development. In: *All About Agile*. 11 June 2007 [viewed 19 April 2016]. Available from:
<http://www.allaboutagile.com/10-good-reasons-to-do-agile-development/>
62. YOURCOACH., 2010. *S.M.A.R.T. goal setting / SMART / Coaching tools / YourCoach Gent* [viewed 17 April 2016]. Available from:
<http://www.yourcoach.be/en/coaching-tools/smart-goal-setting.php>
63. ZAHORSKY, D., 2016. 5 Elements of a SMART Business Goal. In: *Smart Money Article*. 16 February 2016 [viewed 17 April 2016]. Available from:
<http://sbinformation.about.com/od/businessmanagement/a/businessgoals.htm>

Bibliography

1. ALAM, S., 2012. Waterfall Model Advantages and Disadvantages. In: *Buzzle*. 7 March 2012 [viewed 31 January 2016]. Available from: <http://www.buzzle.com/articles/waterfall-model-advantages-and-disadvantages.html>
2. ANJU, R., 2013. What are the advantages and disadvantages of Agile Methodology? In: *ZyxWare Articles*. 24 May 2013 [viewed 31 January 2016]. Available from: <http://www.zyxware.com/articles/3654/what-are-the-advantages-and-disadvantages-of-agile-methodology>
3. BERRY, S. and THOMAS, R, Ph.D., 2008. *Use Smart Objectives to Focus Goals, Plans and Performance* [viewed 26 January 2016]. Available from: <https://www.projectsmart.co.uk/use-smart-objectives-to-focus-goals-plans-and-performance.php>
4. BYPASSLINK, 2014. *Advantages & Disadvantages of RAD (Rapid Application Development) Model* [viewed 31 January 2016]. Available from: <http://bypasslink.blogspot.co.uk/2014/11/advantages-disadvantages-of-radrapid.html>
5. CANIUSE, n.d. *Figure 1: Media Query Support Table* [digital image] [viewed 30 January 2016]. Available from: <http://caniuse.com/#search=media>
6. CHARLTON, G., 2012. Search box placement: examples and best practices. In: *E-Consultancy*. 13 November 2012 [viewed 30 January 2016]. Available from: <https://econsultancy.com/blog/11095-search-box-placement-examples-and-best-practices/>
7. CHASTAIN, S., 2014. What is Graphics Software? In: *About Tech*. 16 December 2014 [viewed 31 January 2016]. Available from: <http://graphicssoft.about.com/od/findsoftware/a/intrographicssw.htm>
8. CIO, 2011. Project Management Methodologies. In: *CIO News*. 4 October 2011 [viewed 31 January 2016]. Available from: http://www.cio.com.au/article/402904/project_management_methodologies
9. CLEARLEFT, 2005. *What We Do* [viewed 30 January 2016]. Available from: <http://clearleft.com/does>
10. CREATIVE BLOQ, 2013. 7 reasons why mood boards are worth the extra time. In: *Creative Bloq*. 12 August 2013 [viewed 3 February 2016]. Available from: <http://www.creativebloq.com/graphic-design/7-reasons-why-mood-boards-are-worth-extra-time-4132367>
11. DAWSON, C., 2015. The Pros and Cons of Utilizing an Agile Methodology. In: *Near Shore Americas*. 22 January 2015 [viewed 31 January 2016]. Available from: <http://www.nearshoreamericas.com/agile-methodology-advantages->

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

disadvantages/

12. DHALL, S., 2009. *Rapid Application Development (RAD)* [viewed 31 January 2016]. Available from: <http://qtp.blogspot.co.uk/2009/04/rapid-application-development-rad.html>
13. EVKO, T., 2014. You Don't Need JavaScript For That! In: *SitePoint*. 17 September 2014 [viewed 30 January 2016]. Available from: <http://www.sitepoint.com/you-dont-need-javascript-for-that/>
14. FEHER, D., 2013. *What are the pros and cons of the Waterfall and Agile Scrum Project Management Approach?* [viewed 31 January 2016]. Available from: <https://www.quora.com/What-are-the-pros-and-cons-of-the-waterfall-and-agile-scrum-project-management-approach>
15. FROEHLICH, A., 2015. Should IT Go Agile? The Pros And Cons. In: *InformationWeek*. 10 June 2015 [viewed 31 January 2015]. Available from: http://www.informationweek.com/infrastructure/pc-and-servers/should-it-go-agile-the-pros-and-cons/d/d-id/1322448?image_number=9
16. GARSIEL, T. and IRISH, P., 2011. *How Browsers Work: Behind the scenes of modern web browsers*. Online: HTML5Rocks.
17. HANKING, D., MURRAY, R. and TIJSSEN, T., 2012. *Rapid Application Development – RAD* [31 January 2016]. Available from: https://prezi.com/tfmpp3mjrjm/_rapid-application-development-rad/
18. HARRIN, E., 2015. The Definitive Guide to Project Success Criteria. In: *GirlsGuideToPM*. 26 January 2015 [viewed 31 January 2016]. Available from: <http://www.girlsguidetopm.com/2015/01/the-definitive-guide-to-project-success-criteria/>
19. HAUNTS, S., 2014. Advantages and Disadvantages of Agile Software Development. In: *Stephen Haunts Blog*. 19 December 2014 [viewed 31 January 2016]. Available from: <https://stephenhaunts.com/2014/12/19/advantages-and-disadvantages-of-agile-software-development/>
20. HENRY, A., 2014. Five Best Text Editors. In: *LifeHacker*. 20 April 2014 [viewed 30 January 2016]. Available from: <http://lifehacker.com/five-best-text-editors-1564907215>
21. HTML.NET., 2010. *Lesson 1: What is CSS?*entutorial - HTML.net [viewed 30 January 2016]. Available from: <http://html.net/tutorials/css/lesson1.php>
22. INDO-SAKURA, 2015. *Rapid Application Development* [viewed 31 January 2016]. Available from: <http://web.archive.org/web/20150115195814/http://www.indosakura.com/en/index.php/sample-page/it-services/development-model/rapid-application-development/>

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

23. ISTQB EXAM CERTIFICATION, 2012. *Waterfall Methodology Diagram* [digital image] [31 January 2016]. Available from:
<http://istqbexamcertification.com/wp-content/uploads/2012/01/Waterfall-model.jpg>
24. LAZARIS, L., 2009. Horizontal Navigation Menus: Trends, Patterns And Best Practices. In: *Horizontal Navigation Menus: Trends, Patterns And Best Practices*. 7 September 2009 [viewed 30 January 2016]. Available from:
<https://www.smashingmagazine.com/2009/09/horizontal-navigation-menus-trends-patterns-and-best-practices/>
25. MAGAR, P., 2013. Advantages of Rapid Application Development. In: *Buzzle*. 1 February 2013 [viewed 31 January 2016]. Available from:
<http://www.buzzle.com/articles/advantages-of-rapid-application-development.html>
26. MALLELA, R. H., 2014. Advantages and Disadvantages of RAD. In: *Scribd*. 30 April 2014 [viewed 31 January 2016]. Available from:
<http://www.scribd.com/doc/221190665/Advantages-and-Disadvantages-of-RAD#scribd>
27. MARHSALL, J., 1997. *HTML Made Really Easy* [viewed 30 January 2016]. Available from: <https://www.jmarshall.com/easy/html/#whatis>
28. MASERANG, S., 2002. *Project Management: Tools & Techniques* [viewed 31 January 2016]. Available from:
http://www.umsl.edu/~sauterv/analysis/488_f02_papers/ProjMgmt.html
29. MIKOLUK, K., 2013. Agile Vs. Waterfall: Evaluating The Pros And Cons. In: *uDemy Blog*. 9 September 2013 [viewed 31 January 2015]. Available from:
<https://blog.udemy.com/agile-vs-waterfall/>
30. MINDUTOPIA, 2013. Horizontal vs. Vertical Navigation – What's Better? In: *Mindutopia*. 27 August 2013 [viewed 30 January 2016]. Available from:
<http://mindutopia.com/2013/08/vertical-vs-horizontal-navigation-whats-better/>
31. MY PM EXPERT, 2009. *The Advantages and Disadvantages of Waterfall Methodology* [viewed 31 January 2016]. Available from:
http://web.archive.org/web/*/http://www.my-project-management-expert.com/the-advantages-and-disadvantages-of-waterfall-software-development.html
32. MY PM EXPERT, 2009. *The Advantages and Disadvantages of Agile Software Development* [viewed 31 January 2016]. Available from: <http://www.my-project-management-expert.com/the-advantages-and-disadvantages-of-agile-software-development.html>
33. NAVEEN, 2015. *What is Waterfall Model in software testing and what are advantages and disadvantages of Waterfall Model?* [viewed 31 January 2016]. Available from: <http://testingfreak.com/waterfall-model-software-testing->

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

advantages-disadvantages-waterfall-model/

34. NAVEEN, 2015. *What is RAD Model in software testing and what are the advantages and disadvantages of RAD Model* [viewed 31 January 2016]. Available from: <http://testingfreak.com/rad-model-software-testing-advantages-disadvantages-rad-model/>
35. NET SOLUTIONS, 2000. *Distributed Agile Development* [viewed 31 January 2016]. Available from: <http://www.netsolutionsindia.com/distributed-agile>
36. PATE, E., 2013. Aims and Objectives of a Research Proposal. In: *Synonym Classroom – Higher Education Prep.* 18 October 2013 [viewed 26 January 2016]. Available from: <http://classroom.synonym.com/aims-objectives-research-proposal-4168.html>
37. PETIT, N., 2014. The 2014 Guide to Responsive Web Design. In: *TreeHouse Blog*. 2 June 2014 [viewed 30 January 2016]. Available from: <http://blog.teamtreehouse.com/modern-field-guide-responsive-web-design>
38. RAPID PROGRAMMING, 2013. *What are the drawbacks of Waterfall Model?* [viewed 31 January 2016]. Available from: <http://www.rapidprogramming.com/questions-answers/What-are-the-drawbacks-of-Waterfall-Model--782>
39. RARELOOP, 2015. *Open Device Lab Southampton* [viewed 26 January 2016]. Available from: <http://odl.rareloop.com/>
40. RASMUSSEN, J., 2015. *What is Agile?* [viewed 31 January 2016]. Available from: <http://www.agilenutshell.com/>
41. RYAN, V., 2009. *Product Review* [viewed 29 January 2016]. Available from: <http://www.technologystudent.com/despro2/prodrev1.html>
42. SARANGI, I., 2014. The Importance of Time Management in Projects. In: *Wiziq Blog*. 21 August 2014 [viewed 31 January 2016]. Available from: <http://www.wiziq.com/blog/importance-of-time-management-in-projects/>
43. SCHOLARSHIP POSITIONS, 2012. Importance of Research Studies. In: *Career Articles*. 29 September 2012 [viewed 29 January 2016]. Available from: <http://scholarship-positions.com/blog/importance-of-research-studies/201209/>
44. SEARCHENGINEPEOPLE, 2010. *Figure 2: Pie Chart of Users with JavaScript Disabled* [viewed 30 January 2010]. Available from: <http://www.searchenginepeople.com/wp-content/uploads/2010/12/image14.png>
45. SEGUE TECHNOLOGIES, 2015. 8 Benefits of Agile Software Development. In: *Segue Technologies*. 15 August 2015 [viewed 31 January 2016]. Available from: <http://www.seguetech.com/blog/2013/04/12/8-benefits-of-agile-software->

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

development

46. SOCIAL ENTERPRISE WORKS, 2008. *How to understand your customers' needs and expectations* [viewed 26 January 2016]. Available from: http://www.socialenterpriseworks.org/wp-content/uploads/2008/03/how_to_understand_your_customer_needs_and_expectations.pdf
47. SPARROW, P., 2011. Waterfall Model: Advantages and Disadvantages of Waterfall Model. In: *iAnswer4U*. 1 November 2011 [viewed 31 January 2016]. Available from: <http://www.ianswer4u.com/2011/11/advantages-and-disadvantages-of.html#axzz3yow1OJVn>
48. TECHOPEDIA, 2011. *What is Rapid Application Development (RAD)?* [viewed 31 January 2016]. Available from: http://web.archive.org/web/*/https://www.techopedia.com/definition/3982/rapid-application-development-rad
49. TECH TARGET, 2010. *What is Waterfall Model?* [viewed 31 January 2016]. Available from: <http://searchsoftwarequality.techtarget.com/definition/waterfall-model>
50. THAKUR, D., 2013. *Rapid Application Development (RAD) Model and its Advantages and Disadvantages of RAD Model* [viewed 31 January 2016]. Available from: <http://ecomputernotes.com/software-engineering/rapid-application-development>
51. THE HTC TEAM, 2013. 7 Web Design Conventions You Should Never Break. In: *High Touch Communications Inc. Blog*. 3 September 2013 [viewed 30 January 2016]. Available from: <http://blog.htc.ca/2013/09/03/web-design-conventions/>
52. TILLOO, R., 2013. *Waterfall Model: What Is It? Its Advantages & Disadvantages* [viewed 31 January 2016]. Available from: <http://www.slideshare.net/rahultiloo/water-fall-model-22606242>
53. TOMAT, A., 2015. *Rareloop Assignment Live Brief* [viewed 31 January 2016]. Available from: <http://mycourse.solent.ac.uk/mod/resource/view.php?id=452229>
54. TUTORIALS POINT, 2008. *JavaScript Overview* [viewed 30 January 2016]. Available from: http://www.tutorialspoint.com/javascript/javascript_overview.htm
55. VERMA, P, E., 2015. Waterfall Model: Design Phase, advantages disadvantages and application. In: *Yuvayana*. 11 May 2015 [viewed 31 January 2016]. Available from: <http://er.yuvayana.org/waterfall-model-design-phase-advantages-disadvantages-and-applications/#disadvantage>
56. VERMA, P, E., 2015. SDLC Agile model design phase, applications, pros, cons, advantages and disadvantages. In: *Yuvayana*. 24 May 2015 [viewed 31 January 2016]. Available from: <http://er.yuvayana.org/SDLC-Agile-model-design-phase-applications-pros-cons-advantages-disadvantages/#disadvantage>

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

- 2016]. Available from: <http://er.yuvayana.org/sdlc-agile-model-advantages-and-disadvantages-with-applications/>
57. WALTER, S., 2012. Responsive Web Design: Layouts and Media Queries. In: *One Trax Pixel*. 23 April 2012 [viewed 30 January 2016]. Available from: <http://www.onextrapixel.com/2012/04/23/responsive-web-design-layouts-and-media-queries/>
58. WE ARE BASE, 2008. *About Base* [viewed 30 January 2016]. Available from: <https://wearebase.com/about/>
59. WELLER, N., 2015. The 11 Best Code Editors Available in 2015. In: *Elegant Themes Blog*. 27 February 2015 [viewed 30 January 2016]. Available from: <https://www.elegantthemes.com/blog/resources/the-11-best-code-editors-available-in-2015>
60. WEMBLER, S., 2002. *Choose the Right Software Method for the Job* [viewed 30 Januar 2016]. Available from: <http://www.agiledata.org/essays/differentStrategies.html>
61. WYSTER, 2008. Using Rapid Application Development for your Software Project. In: *The Wysteria – SYSTEM ANALYSIS*. 28 September 2008 [viewed 31 January 2016]. Available from: <http://wysterdesir.com/2008/09/28/using-rapid-application-development-for-your-software-project/>

Appendices

Appendix A: Importance of Software Testing



Software Testing

Planit Test Management Solutions
Phone: +612 9464 0600
Email:info@planit.net.au
Website:<http://www.planit.net.au>

[What is Software Testing? Why is Software Testing important? We need testing help](#)

Why is Software Testing Important to a business?

Most of us have had an experience with software that did not work as expected. Software that does not work can have a large impact on an organisation. It can lead to many problems including:

- Loss of money – this can include losing customers right through to financial penalties for non-compliance to legal requirements
- Loss of time – this can be caused by transactions taking a long time to process but can include staff not being able to work due to a fault or failure
- Damage to business reputation – if an organisation is unable to provide service to their customers due to software problems then the customers will lose confidence or faith in this organisation (and probably take their business elsewhere)
- Injury or death – It might sound dramatic but some safety-critical systems could result in injuries or deaths if they don't work properly (e.g. flight traffic control software)

Want to know more about how software testing can help your business? Contact [Planit](#).

Related Links

- [What is Software Testing?](#)
- [Why is Software Testing important?](#)
- [We need software testing help](#)

[Software Testing Training](#) | [Software Testing jobs](#) | [Software Testing courses](#)

Appendix B: Original Project Proposal

Project Title

Can a Web Interface improve the quality, reliability and accessibility of an Attendance Tracking system for an Organisation?

Introduction

According to Brune Park Community School located in Gosport, Hampshire, the current attendance tracking system incorporated into their computer systems is outdated and displays a lack of modern functionality in terms of user experience and cross-platform support.

Rational

The key point of this project is to develop skills around further complicated, server side programming languages as opposed to simple PHP. Although this is being conveyed to students by the regular curriculum standards, the desire to push my knowledge to another level beyond the expectations currently set by the educational system will ensure that the end result of the project will be met; a high end, easy-to-use, web based attendance software that can be accessed through multiple platforms and devices.

Aim

To build personal programming and server development skills to manufacture an Attendance Tracking software that is capable of use in an organisation environment.

Objectives

The aim of this project is to address the presented problem by creating a web-based interface that is...

- Easily accessible and usable across multiple platforms that, holds strategic reliability through the use of a sturdy, secure database structure.
- Fluid in terms of design to allow for easy changes through CSS/SASS/SCSS if colour changes or content changes need to be implemented.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

- Responsive to any device it is accessed by, caters to all screen resolutions.
- Personal from a student and faculty point of view.
- Able to bridge the gap between Student and Teacher support, keeping attendance and student resources all under one roof.

Intended Audience

The target audience of the final project software integration will primarily be Brune Park Community School, although there will be the ability of Installation at which an Administrator can be created to add classes/courses, faculty members, student accounts etc. The benefits of this system is that it can be incorporated into any organisation in need of an Attendance Tracking tool, not only educational-based businesses.

Resources

The resources of this project are key factors of the final piece of software that will be developed; the planned components of the project are as follows...

- Web-Based Programming Languages
 - Ruby on Rails (Ruby running through a web server)
 - HTML5
 - CSS/SASS/SCSS
 - JavaScript (Heavily dependant on jQuery)
 - JSON
- Secure Database Possibilities
 - PostGre SQL
 - MySQL
- Open Source Packages
 - Font-Awesome
 - Bootstrap
 - jQuery
- Software used for Development
 - Sublime Text
 - Adobe Photoshop

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Combined together, these components should supply a successful answer to the project's core question, while also catering towards the software development. Without these components, the software would not be able to be created.

Appendix C: Discussions Between Thomas Alderson and Claire Alderson via E-mail About Project Features

Specific details within each e-mail are covered with a black square in order to maintain the privacy of the e-mails and/or personal information of the recipient.

Thomas Alderson Today at 14:52
To: Claire Alderson Sent - TomAldy 
Final Year Project Feature Discussion 

Hi there,

I am writing to you to enquire as to whether you have any idea regarding the attendance tracking system for the organisation that controls your work place?

I would really appreciate your help in this matter to help develop a group of aims and objectives that could cater towards the production cycle of the project.

Many thanks,
Thomas Alderson

Claire Alderson Today at 14:54
To: Thomas Alderson
RE: Final Year Project Feature Discussion 

Hi there Thomas,

Yes of course I am willing to help you in your Final Year Project and your development of the attendance tracking system that you're developing based around Brune Park Community College.

What kind of information do you need?

Thanks,
Claire Alderson

[See More from Thomas Alderson](#)

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Thomas Alderson
To: Claire Alderson
Re: Final Year Project Feature Discussion

Today at 14:57
Sent - TomAldy 



Hi Claire,
Great! Thanks so much for your help in this segment of my project.

I just need an overall view on the current system you have installed and the features you'd like to see in my personally developed attendance system?

The detail doesn't need to be astronomical in size, just a few notes would be fine with a summarising paragraph of the current system leading the e-mail, if that's not too much trouble?

Many thanks,
Thomas Alderson

[See More from Claire Alderson](#)

Claire Alderson
To: Thomas Alderson
RE: Final Year Project Feature Discussion

Today at 15:03



Hi Tom,
That's no problem!

The current system is all an internal structure and can be accessed through the Moodle computer or laptop provided by the school, along with a login and password holding your own files etc. We use a program known as SIMS to control the students attendance, but currently that appears to be working perfectly fine, but there is currently no available platform to cater for staff and pupils to interact with one another, we could use Moodle to do this in the Virtual Learning Environment it comes with, but the design is very off-putting and even discussing this being changed with IT raises a million and one questions that no one wants to answer or solve.

The main headings I'd expect the application to have would be...

Dashboard?
Messaging Service to interact between teacher to teacher?
Classes/Courses?
Timetables? No need to be complicated, just the ability to edit your own personal timetable and allow others to see what you're up to on the current day?
Managers? To find out who to access for contact if the person is not available?

Those are the current ideas I have, I know they're only headings but I am sure you can build on them with your clever computer knowledge!

If you need anything else, don't hesitate to send me a message.

Thanks,
Claire Alderson

[See More from Thomas Alderson](#)

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Thomas Alderson

To: Claire Alderson

Re: Final Year Project Feature Discussion

Today at 15:04

Sent - TomAldy

TA

Hi Claire,

Thank you so much for these pointers, they've helped me with the Site Map already and a plan of features to add to the project once production on the design is completed.

Thanks again,
Thomas Alderson

[See More from Claire Alderson](#)

Appendix D: Objectives in SMART Format

1) Secured accessibility, using login and high level encrypted passwords.

S <i>Specific</i>	An internal network bound application must be secured correctly, without the correct level of encryption, any user could have access to the application, therefore making this objective valid and specific to the project.
M <i>Measurable</i>	The level of encryption is the measurable unit within this objective and can be achieved through specific techniques, although it must be high and irreversible encryption.
A <i>Achievable</i>	The achievable aspect of this objective is created through the use of existing techniques available of encrypting passwords, if it has been achieved in the past, it can be achieved now.
R <i>Realistic</i>	The objective above is certainly realistic, by being a simple objective, it does not hold a determination based on any factors other than the developers ability to achieve the desired outcome.
T <i>Time</i>	There are no time constraints situated within the objective itself, but this is intended to be covered using a separate methodology of a Gantt Chart at a later sector of the report.

2) Accessible Web Interface, scalable to be accessed on any device for portable usage when connected to network controlling application.

S <i>Specific</i>	As this is intended to be a web application, allowing the opportunity to have it available on multiple devices is a constant battle for any web designs, thus rendering this objective specific to the web development industry that the project is built around.
M <i>Measurable</i>	The amount of devices that are used to test the interface are the measurable unit within this objective.
A <i>Achievable</i>	With all of the available tools that have been created to aid web developers in achieving responsiveness with their designs, this objective is certainly achievable through using the correct tools and techniques.
R <i>Realistic</i>	With the web being an ever-growing accessible platform, users are accessing it with more and more devices, thus allowing this objective to be realistic when being accessed via multiple devices, this helps in terms of portability usage while performing practical activities, such as physical education.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

T <i>Time</i>	<p>There are no time constraints situated within the objective itself, but this is intended to be covered using a separate methodology of a Gantt Chart at a later sector of the report.</p>
-------------------------	--

- 3) Sustainable system with a maximum of 200 user accounts and regular connections.

S <i>Specific</i>	<p>Due to this system being held within an internal network and being used on a regular basis, the needs for setting a user limit are certainly important, as any possibility of overload would need to result in a large size of server to handle traffic in the application.</p>
M <i>Measurable</i>	<p>Measuring the amount of users and setting a limit to this gives a use of calculation into measuring how many users the system can hold on a small network, thus being able to discover how many users the system can cater for based purely on network/server size and technology capabilities.</p>
A <i>Achievable</i>	<p>This objective is achievable due to the option of hosting the application on a large network, due to development taking place on a small network, testing is limited, although calculations can be made through testing to create a bases on how many users and connections the application can hold.</p>
R <i>Realistic</i>	<p>As the application is being designed to cater for multiple systems, but as a niche be available to a specific educational organization, the realistic level of this objective is enhanced.</p>
T <i>Time</i>	<p>There are no time constraints situated within the objective itself, but this is intended to be covered using a separate methodology of a Gantt Chart at a later sector of the report.</p>

- 4) Primary support for top five most used browsers, judging by Global Web Statistics.

S <i>Specific</i>	<p>As the web is growing, as are the available browsers used to display websites and due to the embedded rules that each web browser contains, websites can sometimes be displayed differently, making this a specific objective to the overall task.</p>
M <i>Measurable</i>	<p>By using the top five browsers specific by Global Web Statistics, the measurable testing phase can be achieved through the testing phase when measuring the level of success on each browser and whether the objective has been met or not.</p>
A <i>Achievable</i>	<p>The achievable aspect in this objective is reached through the statistics given by websites such as CanIUse.com, which matches up the available support for specific styling settings and features that websites can use and whether each browser supports it or not.</p>

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

R <i>Realistic</i>	Websites face this issue on a regular daily basis, this objective created realism in the task by bringing the testing phase of browsers into the project, rather than focusing on one browser primarily.
T <i>Time</i>	There are no time constraints situated within the objective itself, but this is intended to be covered using a separate methodology of a Gantt Chart at a later sector of the report.

Appendix E: Project Review Documentation



Project Review

CAN A WEB INTERFACE IMPROVE THE QUALITY,
RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE
TRACKING SYSTEM FOR AN ORGANISATION?

THOMAS ALDERSON

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Table of Contents

Background & Introduction	7
Background	7
Project Recap.....	7
Introduction	7
Refined Project Specification.....	8
Previous Development of Specification	8
Why refine the project specification?	8
Updated Project Specification	9
Measurements of Success	11
Values to Measure.....	11
CPU Strain	11
Brute Force Attempts.....	11
Browser Support	12
User Limitation.....	12
Methodology and Options Analysis	13
What is Agile?	13
Why use Agile?.....	13
Key Benefits of Agile	14
Client Involvement	14
Quality	14
Agility.....	15
Project Progress	16
Requirements.....	16
Project Proposal Creation.....	17
Rationale Research	17
Discussions with Brune Park Community School	18
Literature Research	18
Existing Attendance Systems	19
Technologies	19
Software Research	20
Hardware Research.....	21
Architecture & Design	21
Development	23

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Collecting Resources	23
Server Setup	24
Open-Source Assets	24
Pseudo Code	26
Fluid Modal	26
Hardcode	29
Testing & Feedback	30
Project Planning	31
Project Monitoring	33
Project Control	35
Conclusions & Recommendations	36
References	37
Bibliography	40
Appendices	42
Appendix A: Project Proposal	42
Appendix B: Project Specification	45
Specification	45
Description	45
Features	45
External Interface Requirements	45
i. User Interface	45
a. Follow conventions of standard Dashboard styles & templates.....	45
ii. Hardware	46
iii. Software	46
Other Non-functional Requirements	46
i. Performance	46
a. The speed of the system is based entirely on the device that the software is performing on. This is why a baseline has been specified above in the Hardware requirements.....	46
ii. Safety	46
a. Health and safety terms are based purely on the policies written by the business using the product.....	46
iii. Security	47
a. Data Protection Act Compliance.	47

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Appendix C: Refined Project Specification	0
Project Title	0
Project Aim & Objectives.....	0
Aim.....	0
Objectives	0
Project Interface Map Scope	1
Features.....	1
External Interface Requirements.....	2
i. User Interface	2
ii. Hardware	2
iii. Software	2
Other Non-functional Requirements.....	3
i. Performance.....	3
a. The speed of the system is based entirely on the device that the software is performing on. This is why a baseline has been specified above in the Hardware requirements.....	3
ii. Safety.....	3
a. Health and safety terms are based purely on the policies written by the business using the product.....	3
iii. Security.....	3
a. Data Protection Act Compliance.....	3
Appendix D: Case Studies Article on Agile Methodology Examples	4
Appendix E: Notes from Meeting with Head of Attendance Systems of Brune Park Community School	8
Appendix F: Technological Resources being used to run the Application.....	9
Resources.....	9
Technological.....	9
i. Ruby on Rails Framework	9
ii. Networking	10
iii. Web Development Software	10
iv. Database Management Software.....	10
v. Web Server.....	11
Appendix G: Low Fidelity Design Attempts	12
Design One	12
Design Two	13
Appendix H: Demo Version of Admin LTE2 Design	14
Appendix I: Commands Used to Create Application Server and Database	16

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Appendix J: Steps to Installing Admin LTE2 on Ruby on Rails Application.....	17
Appendix K: Bootstrap Modal Documentation	21
Live Demo	23
Appendix L: Hardcoding Evidence, page designs with accompanying code.....	24
Users Page	24
Full Page	24
Code	24
Timetable Demo Page with Evidence of Booking Feature.....	31
Full Page	31
Code	32
Appendix M: Time Management Based Contingency Plan	37
Appendix N: Updated Gantt Chart with included Minutes	38
Appendix O: Skeleton of Final Project Report	40

Background & Introduction

Background

The purpose of this report is to evaluate/review the journey of development within a specific project. Throughout this report, the stage that has been currently met in the project will be discussed and analysed on how successful the project production process has been up until the stage that has been reached.

Project Recap

Recapping the project will give an overview of the current expected conditions of the project outcome, recapping the project will give a further understanding of why any changes may need to be implemented as a result of beginning the production evolution, as well as a reference to justify these changes.

Please see Appendix A for the Project Proposal, created to give an overview of the project with its aims, objectives, resources and expected outcomes.

Introduction

This report will review and evaluate the progression of the final year project titled "Can a Web Interface Improve the Quality and Accessibility of an Attendance Tracking System?" and how the advancements of the production and planning process have been followed successfully, along with the positive and negative outcome of each task with its domino-effect on future developments.

The project production level has reached the implementation level, at which the design has been finalised, with programming well underway, the production sector of the project has accelerated beyond the expected level, this is to certify that there is enough time for correct damage control if any unexpected problems are to occur.

Refined Project Specification

"Cutting corners in project planning is a recipe for disaster, no matter what the reason. The initiation phase is critical to the success of the project as it establishes its core foundations. Effective project planning takes into consideration all aspects of planning including stakeholder engagement, benefits mapping, risk assessment, as well as the actual plan (schedule) itself." **(Priority 2013)**

Previous Development of Specification

In past documentation dedicated to this project, the project proposal combined with the created specification, although this specification was not in much detail as it was planned on being revisited within this review report. Within the previous specification, there was a lack of depth which will be built upon, much like building a piece of architecture, a base must be created to allow for a solid structure to be constructed upon.

Please see Appendix B for the current development of the Project Specification, produced in the Project Definition stages of the project.

Why refine the project specification?

"Continually refining the project's scope allows people working on the project to think about how that project fits in with the concerns of the business as a whole." **(Morley 2012)** Creating a refined project specification will ensure that thorough details are further understood to follow through while project production takes place, a specification can define measurable values that can form the testing period of the project development. Within the project testing phases, the project specification will be revisited as it will contain key figures, whether qualitative or quantitative to help measure the outcome of the project and whether a sufficient

level has been met to consider the project a success, if not, the project production sector must be revisited until success is reached.

Updated Project Specification

The project specification will now be 'fine-tuned' after the project production has begun, as there are now added values that need to be taken into account in the specification, as the final product is intended for use of an educational organisation, they need to be handed a document containing all of the information on what the product can do/intends to do, this is where the specification comes in.

Please see Appendix C for the fully detailed Refined Project Specification.

Specific changes have been made within the newly refined Project Specification, there has been a change in the Aims in that they are more refined to the project, as opposed to being specified to the users needs and being very vague.

A very noticeable change is in the functionality measurements, with a cap added to the objectives of being able to support a maximum 200 staff members, this way the system is not strained and there will also be a limitation built into the system.

Although, there has also been a change of objectives to include the support of web browsers that can be used, using data provided by the Global Web Statistics (w3Counter) to justify which web browsers will be supported over any others.

According to Geng, browser support is a very important aspect of any web interface, even back in 2010, so of course, the problem has only multiplied since...

"Nowadays everyone's using a different browser. Between popular options like Firefox, Safari, Chrome and Internet Explorer, which make up close to 98% of the internet market share of browsers, and the other little known browsers like Konqueror, there are a multitude of browsers being used to view your site." **(Geng 2010)**

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Sustainability of high security has also been taken into account, with an added objective of ensuring that passwords are encrypted to the correct standards that are up to date with todays expectations of protection when creating user accessible accounts. Pornin, in a Security oriented blog, back in 2011 described the importance of why a password should be encrypted/hashed...

"A first thing to note is that there are many people who talk about encrypted passwords but really mean hashed passwords. An encrypted password is like anything else which has been encrypted: it has been rendered unreadable through a process which used an extra piece of secret data (the key) and which can be reversed with knowledge of the same key (or of a distinct, mathematically related key, in the case of asymmetric encryption)." (Pornin 2011)

A re-worded but still evident objective that has been redefined is the scalable factor of the design, so that it can be accessed on any platform or device, keeping to the constraints of browser support, this is discussed brilliant by Rampton 2014 in an article all about the importance of responsive design.

An addition of the interface map scope has also been added to show the layout of the application would be in terms of following a map of the system down from login, all the way through to logout, this is helpful in production stages as a measurement of completion. Why create a site map? This is brilliant explained by Koht, an active web project artisan...

"Whenever we engage in a web production, we always build a site map. A site map is the architectural framework for the project, whether a large scale system or a five-page website. When clients understand how to build site maps, the web design process becomes increasingly efficient for both parties." (Koht 2010)

Measurements of Success

"Creating good, solid measures is the first step to developing your performance scorecard. Measures are quantifiable performance statements, and they must follow certain guidelines." (Olsen 2011) The measurements of success that need to be applied to the project are very important to regarding the level of completion that has been met. Due to using the Agile methodology, the success measurement document will consistently be referred to in order to help decide whether or not the next stage of production can begin, if success tests fail, then the stage of production must be repeated until the criteria specified by the measurements of success dictates that significant completion has been met.

Values to Measure

CPU Strain

As discussed by Barett 2012; The strain of the system is an important benefactor, this is usually measured through the use of monitoring the central processing unit using a piece of software known as Activity Monitor (for a mac) and Task Manager (for Windows computers) – these are the intended operating systems that the server will be running on. The expected minimum usage the program should use of CPU usage should be no more than 15%, and even this number is rather high for the particular application that is running on the server.

Brute Force Attempts

A basic understanding of a brute force attack...

"A common threat Web developers face is a password-guessing attack known as a brute-force attack. A brute-force attack is an attempt to discover a password by systematically trying every possible combination of letters,

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

numbers, and symbols until you discover the one correct combination that works. If your Web site requires user authentication, you are a good target for a brute-force attack." (**Virginia College 2006**)

Due to the password needing to be highly protected and encrypted to the correct level, a measurement indicator of how secure the password encryption is will be tested through the use of a brute force attack program. This should fail and not allow a user into the application due to how solid the encryption is. This is a simple Boolean test on whether it is true that security is high enough, or false.

Browser Support

Supporting web browsers is an ever growing issue for any web application developer, due to the support that individual web browsers offer when it comes to specific web-kits included and the different ways in which some browsers read code, error proofing levels etc., as discussed by **CrossBrowserTesting 2010** in a discussion of browser support and testing; it is best to keep this down to only five browsers, this way success can be measured on whether the design matches the top five browsers supplied by w3counter's data.

User Limitation

Limiting the amount of users at 200 accounts is very important to the integrity of the server, of course there are particular circumstances which can allow the system to withstand more storage of user accounts and their data, such as running on a server which a much higher memory capacity, in terms of random access memory and hard disk memory.

Although, measuring that the system is able to cut off user creation at 200 user accounts is the measurable factor here, as anymore users could possibly create lag on the server processing speed and will slow the entire application down for all users that are accessing the application at one time.

Methodology and Options Analysis

The chosen Methodology that is being followed throughout the project's production is Agile, as stated in the Definition report, although Agile is built for use in team project management, the developer of this project is a perfectionist and will certainly re-iterate over each phase of the project production until perfect standards are met as stated in the planning documentation.

What is Agile?

Agile is a methodology for a speedy developer, it consists primarily of working sprints; "In product development, a sprint is a set period of time during which specific work has to be completed and made ready for review." (Rouse 2011) which are used in order to complete work at a rapid pace, due to the speed at which the sprints usually are; Agile is built for a team of developers rather than an individual taking on an entire project alone.

Why use Agile?

Due to the nature of the project, along with other commitments the developer has to stick to, Agile seems the best approach in achieving the desired result of the project in a controlled manor. As described by Bhatia 2013, Agile focuses on achieving particular sectors of the project, whether it be planning, production, testing or implementation and allows the developer(s) to continuously reiterate over the current task that is being focused on until satisfactory criteria is reached.

Please see Appendix D for four case studies discussed in an article about Agile, two success stories and two failure stories.

Key Benefits of Agile

The ideology of Agile is to ensure that time constraints are successfully met, whilst also allowing all team members to continuously work on a task until it's 100% completion.

Client Involvement

"Agile development principles encourage active 'user' involvement throughout the product's development and a very cooperative collaborative approach. This provides excellent visibility for key stakeholders, both of the project's progress and of the product itself, which in turn helps to ensure that expectations are effectively managed." (**Waters 2007**)

Due to the development of the Attendance System being overviewed by Brune Park Community School, using Agile allows the staff at Brune Park School to continuously view the creation of the entire application, meaning if any necessary changes need to be made, whether big or small, they easily can be due to the project being completed in small steps/sprints. Upon finishing a task, the client can view the project development at this stage and give feedback of what can be changed or whether the needs have successfully been met.

Quality

"By breaking down the project into manageable units, the project team can focus on high-quality development, testing, and collaboration. Also, by producing frequent builds and conducting testing and reviews during each iteration, quality is improved by finding and fixing defects quickly and identifying expectation mismatches early." (**Seague Technologies 2015**)

Agile allows for testing of each phase throughout completion, meaning that every aspect and component of the application can be tested straight away to ensure that

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

each sector is completed to the details and expectations provided by Brune Park Community School. Regular testing is one of the biggest bonuses of the Agile methodology, this way the project can run through a 'checklist' manor for the developer; with an order to follow.

Agility

"The main reason why the Agile methods are called "Agile" is because the iterative lifecycle is designed to accommodate change. Work is done in short "iterations" (or Sprints) of only a few weeks, and the transition from one iteration to the next includes taking stock of what may have changed since the iteration began and how to adapt to those changes." (Global Knowledge n.d.)

The key component of the Agile methodology, with flexible properties embedded into the methodology, any particular stage that Brune Park Community School feels is inadequate upon testing, the client can make changes upon the criteria if necessary. Cost is a contributing factor of change from the client, although this can be discussed once a consumer requests the change and can be resolved at such a stage, if it arises.

Project Progress

The progression of the attendance tracking system has been extremely successful, ranging from design level all the way through to starting production processing and thus far there hasn't been any serious issues. This section intends to discuss, analyse and describe the progression that has currently been reached, using illustration techniques to support the points being made. Throughout this section of the report, the structure will follow the key stages of development, showing evidence of what has been achieved and to what level, along with real-time research that has been made during creation to match the specific stage of progression.

Here is a diagram or figure of the specific level that has been reached in the project so far...

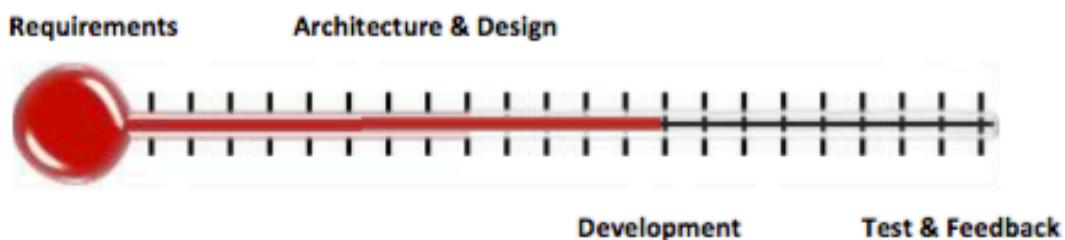


Figure 1: Illustration of Project Development

As you can see by the illustration above, the project is now at development stages, this is the stage at which the production of the final product has begun; meaning that the requirements and architecture & design stages have both come to an end. Below is a break down of each sector of the project with a reflection on how each stage went with accompanying evidence of the completed work.

Requirements

The requirements stage began with a simple task of outlining the task of the project and the intended solution that it needs to create, producing a rationale and collecting necessary documents together to outline the components, resources, apparatus. The requirement stage is the most important level of the project

progression in the production environment, without correct development of basic ideas, they cannot grow into a final product.

Gathering requirements primarily revolved around collecting necessary researching resources in order to build up a project research portfolio that can be referred to as development of the project continues. This stage was built solely around finding facts and figures that have been recorded by others that have built an attendance system in the past, building a small collection of research to add to any evidence as to why, in general terms, specific parts of the project can be treated in such a manor.

Project Proposal Creation

Although this is more specific to the project as a whole, without the project proposal document, there would be nothing to lay out what exactly needs to be researched to let the production process commence. As already discussed in this report, the first draft of the project specification was certainly very vague in terms of what it laid out and explained, along with very 'patchy' objectives and overall aim. To take a look at the updated Project Specification, please see Appendix C.

Rationale Research

Choosing to build a staffing attendance system for a school was not the original idea for the project, but due to development being based around one individual, the workload needed to be cut-down to allow for completion to be possible within the given time limit. To create a rationale to follow for the project to be based around meant that the project became a much more concrete creation, giving it a bigger effect on the overall completion as it must be able to work for an industrial organisation.

The rationale for the project was introduced through a colleague of the project developer, who is employed at the school known as Brune Park Community School, they are having progressively multiplying issues with their attendance system and even their portal that the students can access their resources from. The primary

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

project aim was then created, to replace the existing system that the school already has installed; with a web interface that can be ran on a local server from the school to be accessed from any device once connected to the schools' wireless local area network connection.

Discussions with Brune Park Community School

To really understand and justify the needs and expectations that the school were looking for, a meeting was conducted between the developer and the head of attendance systems in the school. Having a meeting to get the full desired outcome of the project was very important to the project development cycle, notes were taken during the meeting to ensure that correct analysis and understanding can be built around the given discussion.

Please see Appendix E for the meeting notes.

Although, judging by the notes there aren't many points made, they are a perfect baseline for development to begin in terms of architecture and design. The meeting only managed to be twenty minutes long, so the ability to gain a lot of notes after introductions had been made could have been majorly restricted due to time constraints.

Literature Research

Literature research is a necessary, fundamental process of developing any piece of software, a collection of research was made into particular areas of development in order to support the production process of the project. Research was not laid out in any particular order, as they say, knowledge is power, therefore the more research that was collected, the better chance of succeeding in creating a attendance system beneficial to helping Brune Park Community School.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Existing Attendance Systems

There are many attendance systems that exist in the world today, meaning that there needs to be something different in the development of the project attendance system to ensure it serves a purpose; creating a bespoke system seemed suitable for Brune Park Community School, although there are reasons to support them installing a system that already exists.

The table below was created specifically as an overview of research conducted for easy reference whilst development of the attendance system continued.

System	Organising Abilities	Features	Timetabling	Support	Total
Fastrak	7	6	0	8	21
FastVein T&A	8	8	8	8	36
SIMS	7	8	5	3	23
BePunctual	4	3	8	6	21
MyAttendanceTracker	3	1	0	3	7

By using the table above, while working on specific areas of the attendance system, relative comparisons can be made between the highest scoring contenders for that specific aspect of the system to see what can be improved and implemented on the system being produced for Brune Park Community School.

Technologies

Researching necessary topics and resources that will be of use to the project development help to decipher which technologies would be of the best stature to ensure the best performance is met when creating the application. The main goal of the attendance system is to be web-based in terms of an interface, allowing users to access the interface via their web browser through a local area network connection (wireless) depending on what device they're viewing the application with.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

To fully understand the technologies that can be used to create the application, a definition and analysis of the technologies being used was created to fully understand how the technologies work with the benefits and issues of the chosen technologies.

Please see Appendix F containing a list of the technologies being used to run the entire application.

The project proposal contained an entire multitude of software packages that can help maintain the application to, such as Sublime Text, Ruby on Rails, with web programming languages such as HTML5, CSS, JavaScript and jQuery – which are the primary coding languages that the front-end of the application will run on; with the backend using Ruby to process the backend data being fed into the database and server.

Personal experience, along with career experience, was a huge benefactor for the developer in terms of choosing a web interface to produce the application through, due to this experience support; the amount of research that could have been made into the specific technologies was not needed; if any research was needed- it would have happened upon discovering errors, if any occur throughout the production stages of the application.

Software Research

Software research is an important key role to creating the application, without using specific software to help create the application, then there would be no possibility of creating the application at all – not only this; but application support can also cut development time in half, helping cost and time management due to the added features the program(s) have to offer through syntax highlighting, error-handling and more.

The software that would be used to bring the entire application to life are as follows, if these pieces of software do not sound familiar, the application is being created

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

through an Apple Mac computer, not a Windows computer, but the interface can be accessed on either platform if needed...

- Sublime Text
- Terminal
- File Management

Although research was made into many other software packages, due to cost and time management issues that could possibly occur if using higher graded software packages, these programs come stocked with an Apple Mac computer and they also have counterparts on Microsoft Windows and Linux computers, if the system is to be maintained by future individuals.

The only piece of software which does not come stocked on all computer platforms is Sublime Text, but this program is open-source and may be downloaded on any platform – although the files that control the application can certainly be edited in any file-editor of the developer's choice.

Hardware Research

Necessary research into hardware capabilities had to be made, this part of the research and development sector ensures that there is a baseline system that has the ability to run the application with several users accessing it at once, with the server able to process such an amount of requests.

To see the full extent of the hardware capabilities expected to be enough for the server to be maintained on, please see Appendix C, a project specification that goes into details of the requirements of hardware.

Architecture & Design

Architecture & Design was certainly a much more physical development side of the production process of the project, deciding on a design is always a tough stage of any project and throughout this sector, the results are very surprising of what was

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

eventually used as the final design. The particular page that was used as the basis of decided on whether the design was good enough quality was the dashboard, designing the dashboard was an upmost importance as this is the first page that any user is welcomed to upon logging into the system.

To successfully choose a design based upon the dashboard page mock-ups, there were two designs created, both in low fidelity and the decided design would then go on to be designed in high fidelity; although this did not go as planned. As design stages continued, the struggle to create a user-friendly design that was able to match an educational environment was a struggle.

To see the low-fidelity design attempts made, please see Appendix G, although these were not used.

These two designs were not used due to the lack of ideas flowing for the developer, due to this; further research into web application interface design was performed. The research phase of the interface design period led to the discovery of open-source interface design, although if this was a route that would be travelled down to bring the interface to life, there would need to be an integration of using the specific open-source design with a Ruby on Rails framework templating system.

Due to the time being taken up by the designing process, to ensure that the project timeline was left intact and on schedule, the use of an open source design was the only resort possible after deciding not to create a unique design for the interface. After making the decision to use an open source design for the web interface, a suitable layout needed to be found that would easily merge with the Ruby on Rails framework to be able to be edited easily.

After much research and deliberation of whether to add to the cost of the project with purchasing a design or not, the developer discovered Admin LTE 2, a dashboard template available for project and re-sale use, created by Abdullah Almsaeed; an open source dashboard/interface design that has been converted into a gem that

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

can be implemented into Ruby on Rails, as simple as pushing a button to install. The open-source dashboard theme is stocked with components and resources that can support the illustration of the data fed into the application from the database and for the sake of the time management keeping consistent; Admin LTE2 became the choice of design for the interface.

Please see Appendix H to see demo screenshots of the dashboard theme known as Admin LTE2 (the chosen design for the project).

Development

Progression of the development cycle once it had begun took a snowball effect in terms of how quick the application was coming to life, although not fully completed yet; the development stage has certainly been the quickest forming and the most exciting part of the project; bringing the application to life. The development cycle consisted of a three stage plan, using these three steps, the life span of the development stage can fully be understood.

The three stage plan is as follows...

1. Collect Resources
2. Pseudo Code
3. Hardcode

Collecting Resources

This stage of the development process is at which basic templates for coding are created, along with setting up the application baseline level. The resources collection stage simply was a collection of all the open source materials that can be created and setup before true development of an independent application can begin.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Server Setup

Setting up the Ruby on Rails server began the development stage, firstly there needed to be an installation of Rails on the workstations (computers) that the server was being developed through; this was a simple program download and install which would not need to be implemented into a live version as it would be running through a server rather than a local machine.

Once installation of Rails was completed, the command line needed to be used in order to generate the entire application; Ruby on Rails has the ability to create a baseline application for you, which you then cater to your own needs after specifying how you would like the database tables to be. As you can see in **Appendix I**, the collection of commands used to create the application are fairly minimal, this is because Rails simply deals with the processing of data and not how the data is controlled and manipulated, that comes down to the coding in the files of the application, created by the developer.

Open-Source Assets

Collecting and implementing the necessary pieces of software are beneficial to correct house styling's that users are accustomed to when visiting web interface and can keep to the timeline of the project; these are assets such as icon packages and designs that are being implemented.

Font-Awesome

Font-Awesome is an icon package, open source and available to be used in any product, whether being re-sold or simply used for private use. Due to the high usage of the Font-Awesome icon package, users will find themselves accustomed to the icons that will be embedded within the interface, this allows the design to remain open-sourced in every way possible as explained earlier, to keep within the time frame allocated for project completion.

To install Font-Awesome, all that was needed to be added to the application(s) files was a simple line of code to include a cascading stylesheet that will add the

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

necessary classes containing the icons to the design. For evidence of the addition of Font-Awesome, please see the diagram below containing the code and a small collection of icons included in the stylesheet.



```
meta_content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=0" name="viewport">
<link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css" rel="stylesheet" type="text/css" />
<link href="https://code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css" rel="stylesheet" type="text/css" />
<%= stylesheet_link_tag "application", media: 'all', 'data-turbolinks-track' => true %>
```

Figure 2: Evidence of Font-Awesome Stylesheet Addition

Admin LTE2

The installation of Admin LTE2, primarily came from the application itself, there was not much that needed being performed in terms of changing any code; by accessing GitHub, there is a gem created that can install all of the necessary files and folders to ensure that the Admin LTE2 design would wrap the existing content, as of yet in the application, there was none.

Installing the gem known as 'adminlte2-rails', created by Nicolas Bersnard ensured that correct implementation of the Admin LTE2 dashboard design was fully performed correctly.

To see a full breakdown of the steps that were used to install the theme on the application, please see Appendix J.

Pseudo Code

Pseudo code was created initially by the Rails Application to help follow specific standards, although not every feature has yet been developed for the system, the current development stage that has been reached has an accompanying pseudo code for each file that has been created, in this instance, one example will be shown in order to provide evidence of this practise being put into place.

Pseudo code is used to layout the expectations of a feature step-by-step to help a programmer to understand what is happening in a file, thus allowing them to convert the pseudo code from normal English into a programming language, in this case; Ruby or HTML and CSS.

Fluid Modal

Due to the use of the Bootstrap framework coming bundled with the Admin LTE2 design that is being used, there are built in features that can be taken advantage of to help content appear in a more dynamic manner to the user(s) of the application; in this case, Modals are the topic of discussion with progression of how they are being used. To understand what a Modal is, please see Appendix K for a full explanation given in the Bootstrap documentation.

Looking at the way a Modal is created, it is a static piece of content and cannot be physically changed once a page has loaded it, due to the fact it is an element created within the page; to make a modal work with the needs of the application, the modal needed to change depending on if a hyperlink is clicked, the job of a modal will be to load a form in; creating a dynamic fade-in effect for the user(s) of the application.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

This is a rather complicated task when explained in written terms, we just need the content within the modal to change, dependant on what link has been clicked; to achieve this there was the issue of having to use jQuery and rather complicated JavaScript to achieve what we needed; therefore, using pseudo code to layout the steps of the code before fully programming the script was the best approach.

The Pseudo code generated for this task is as follows, although the pseudo code did need to be revisited after error logging...

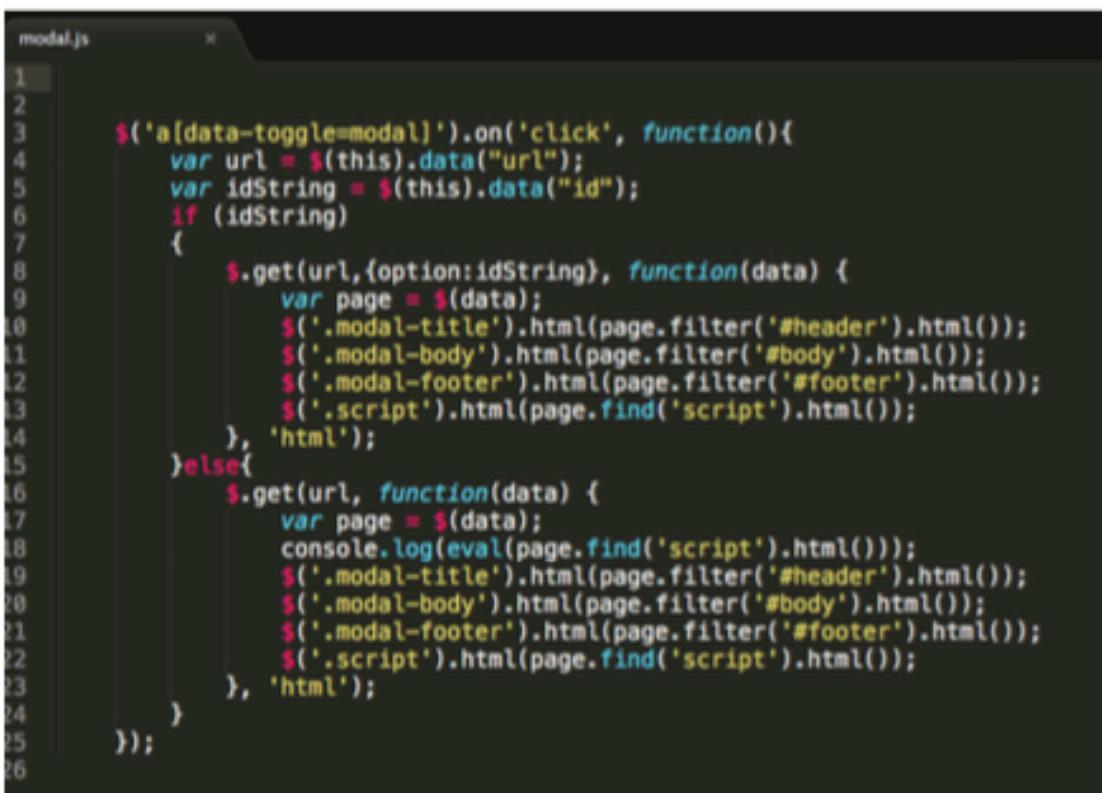
CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

```
Modal is always present within the application's main page.  
1 Modal is always present within the application's main page.  
2 We need to change the content of the modal on the page, rather than it being  
blank.  
3  
4 Pseudo Code for above task...  
5  
6 [—————]  
7  
8 Click on Specific Link.  
9 Grab the URL of the link specified.  
10 Call to the page that the URL is requesting using GET functions and finding  
out the HTTP response returned.  
11 If response returned is 200 (OK)  
12 Filter through the returned HTML, finding the header, body and footer.  
13 Then  
14 Inject the returned HTML that has been filtered into the Modal that already  
existed.  
15 If the page doesn't exist Then  
16 The script fails and the modal does not load.  
17  
18 [—————]  
19  
20 Due to this being an application, if a link is clicked that contains a  
particular ID or needs form data passing from one modal to another, we need  
to support this.  
21  
22 Revised Pseudo Code below for new task...  
23  
24 [—————]  
25  
26 Click on Specific Link.  
27 Grab the URL of the link specified.  
28 Check to see if there is any data stored within the link using data  
attributes such as data-id="" Then  
29 Call to the page that the URL is requesting using GET functions and finding  
out the HTTP response returned.  
30 If response returned is 200 (OK)  
31 Filter through the returned HTML, finding the header, body and footer.  
32 Then  
33 Inject the returned HTML that has been filtered into the Modal that already  
existed.  
34 Also inject data supplied with the HTML by using the same filter function as  
previously mentioned.  
35 If the page doesn't exist Then  
36 The script fails and the modal does not load.
```

Figure 3: Modal Pseudo Code

...this code was then able to be condensed once the final, hard coding version had begun to be developed...

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?



```
modal.js
```

```
1
2
3    $('a[data-toggle=modal]').on('click', function(){
4        var url = $(this).data("url");
5        var idString = $(this).data("id");
6        if (idString)
7        {
8            $.get(url,{option:idString}, function(data) {
9                var page = $(data);
10               $('.modal-title').html(page.filter('#header').html());
11               $('.modal-body').html(page.filter('#body').html());
12               $('.modal-footer').html(page.filter('#footer').html());
13               $('.script').html(page.find('script').html());
14            }, 'html');
15        }else{
16            $.get(url, function(data) {
17                var page = $(data);
18                console.log(eval(page.find('script').html()));
19                $('.modal-title').html(page.filter('#header').html());
20                $('.modal-body').html(page.filter('#body').html());
21                $('.modal-footer').html(page.filter('#footer').html());
22                $('.script').html(page.find('script').html());
23            }, 'html');
24        }
25    });
26
```

Figure 4: Modal JavaScript Final Code

This is a simple example of how pseudo coding was used throughout the project, although the hard-coding side was not included or complicated as quickly as it appears, the full code has been included to help view how helpful pseudo code was throughout producing these types of features.

Hardcode

The hardcoded process was completed primarily through the use of the program known as Sublime Text 3, with many features such as support for multiple file skeletons if a folder is dragged into the program; this helped control the start of the development cycle when switching between multiple supporting files if an error had occurred spontaneously.

Due to the hardcoded sector of the project being fairly new, this section will be more of a portfolio that showcases the current level that the project is at with the accompanying code that has created each file. Although full explanation has not

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

been given in the hard coding sector as of yet, there will be a much bigger explanation in future documentation of the project.

The current coding languages implemented into the application are Ruby, HTML5, CSS, JavaScript, jQuery and SQL. There are many coding languages that could also be used but the use of too many programming languages risk the speed of the application rapidly decreasing; this is why the use of Ruby is imminent in the first place.

To view each page currently in development stages, along with the code to accompany it; please see Appendix L.

Although more depth can be discovered in this section, based on the two pages that have been attached in the appendix (User page and Timetabling page) it can clearly be seen that each page is following the same structure due to the layout being established in a totally different folder of its own with the content being loaded based on whatever URL is visited by the user.

The development stages are currently still in progress and full update of the final product will be given in future documentation.

Testing & Feedback

The testing and feedback stage has not been reached as of yet.

Project Planning

Planning the project was a tough to ensure that correct risk mitigation, analysis of resource accessibility and deciding on what techniques would be the best to use to reach the necessary outcome that is intended. Project planning ensured that the necessary documentation was ready to be used when it came to design and development stages, contingency plans and time keeping documents were to follow the project through and be updated if any issues that may occur.

The only contingencies that were taken into account within the project definition upon completion was the issue of time management, most issues, if they were to occur were filtered into risks and how they could be changed. Contingencies are aspects that are out of the project management teams control and time is definitely a factor that cannot be changed.

Please see Appendix M for a created contingency plan based around time management and what to do if any issues are to occur.

The only issue that arose in terms of time management was in the design stages, the designs that were being created did not match the specific requirements or maturity to be used within an educational institute, according to the designer. This took place within the architecture and design phase, although keeping to the time management, the designer had to deviate from the expected plan of designing a unique template, resulting in the use of an open-source design, discussed in the project progress section above.

According to Noyes, a professional in discussing the use of open source software for business projects, discussed the freedom of included with open source software usage...

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

"When businesses turn to open source software, they free themselves from the severe vendor lock-in that can afflict users of proprietary packages. Customers of such vendors are at the mercy of the vendor's vision, requirements, dictates, prices, priorities and timetable, and that limits what they can do with the products they're paying for." (Noyes 2012)

The use of open source software had no reflect upon the expected outcome of the product, although there will need to be necessary notes and copyright notices placed within the application to ensure that correct credit is given to work that was not completed by developers, but is open source software.

Updating a Gantt Chart is extremely important, explained by Anthony 2014 in a blog post discussing the matter.

An updated Gantt Chart has also been created specifically to ensure a detailed up-to-date reference of how development time was spent on the project, this updated Gantt chart now displays the expected amount of minutes each task should/has taken to complete, although this will be updated to hours in a future version in the final documentation of the project, as opposed to being shown in days in the previous version.

Please see Appendix N for the updated Gantt Chart.

Project Monitoring

Monitoring the development of the project is a key aspect in the overall view of the project production, the project was planned out through following the methodology chosen after research was completed; as already explained, the phases of the project plan were as follows...

1. Requirements
2. Architecture & Design
3. Development
4. Testing & Feedback

Following this layout stated by the Agile methodology ensured that the correct tiers were followed to accumulate a successful product by the end of the project lifecycle.

Unexpected delays can also be a giant risk, as discussed by **ITtoolkit 2012** in an article all about project delays and how to deal with them; this document is a great support reference to use in the kind of situations discussed below.

The only issue that arose that was unexpected, as previously mentioned on a number of occasions was the change in the design phase at which no unique design for the product has been created, but instead there has been the decision to use a open source template known as Admin LTE2. The issue that arose was not included or expected to occur, therefore was not present in the contingency plan or risk mitigation document; the solution was purely a last-minute decision by the project development team.

This is where the impact and issues of the choice of using the Agile methodology has appeared, re-iterating over the same phase up until completion appeared to be problematic due to the developer being unable to create a sufficient design. A

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

contingency plan, although small, has now been produced and can be referred to in these types of time constraining situations.

Project Control

Controlling the project is all based around justifying and measuring the level at which the project production is at, using measurable values within the project allows any witnesses of development to understand how long is left in development until project completion. Through the use of project metrics, stated in earlier stages of the project definition, pin-pointing the current level of production is very simple.

The project metrics stated earlier in the definition are as follows...

1. Scheduling through Time Management
2. Productivity through illustrations of effort distribution.
3. Quality based on client feedback.

The project specification has been updated to allow for measurement of testing services in order to ensure that adequate project completion has been reached. The specific baseline values that need to be met by the end of the project are stated in the refined specification above in this document, these can be seen in Appendix C.

Conclusions & Recommendations

In conclusion, the project is progressing at a phenomenal rate and although there have been deviations from original expectations of the project production, especially in the architecture & design phase, these are brilliant learning curves that can help in experiencing true project management. The next step in the line of fire for the project ladder is to continue with development production and follow with a final report full of discussions on the entire project with specific opinions on how the project has been throughout all stages as a whole.

Please see a skeleton that details the layout of the final report of the project production in Appendix O.

A small contingency plan has been created within this document to ensure that any future issues that may occur that can directly inflict time management, due to factors out of the developers control have a 'plan B' to resort to if needed.

References

1. ANTHONY, K., 2014. Gantt Chart Tips: When To Use Milestones in Your Project Plan. In: Smartsheet Blog. 7 January 2014 [viewed 4 January 2016]. Available from: <https://www.smartsheet.com/blog/support-tip-milestones-in-project-management>
2. BARETT, J. T., 2012. The Importance of a Computer CPU [viewed 3 January 2016]. Available from: <http://smallbusiness.chron.com/importance-computer-cpu-54856.html>
3. BHATIA, A., 2013. Agile: Focus on goals, more than methodology. In: ITNews. 2 April 2013 [viewed 3 January 2016]. Available from: <http://www.itnews.com.au/news/agile-focus-on-goals-more-than-methodology-338324>
4. CROSSBROWSETESTING., 2010. Which Browsers Should I Test? [viewed 3 January 2016]. Available from: <https://crossbrowsertesting.com/browsers-to-test>
5. GENG, S., 2010. The Importance of Cross Browser Compatibility: Tips and Resources. In: Boilerplates & Other Tools – Noupe. 30 April 2010 [viewed 2 January 2016]. Available from: <http://www.noupe.com/development/the-importance-of-cross-browser-compatibility-tips-and-resources.html>
6. GLOBAL KNOWLEDGE., n.d. 12 Advantages of Agile Software Development [viewed 4 January 2016]. Available from: https://cs.anu.edu.au/courses/comp3120/public_docs/WP_PM_Advantages_ofAgile.pdf

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

7. ITtoolkit., 2012. Manage Project Delays: Proactive Steps to Minimize the Unexpected [viewed 4 January 2016]. Available from:
<http://www.ittoolkit.com/how-to-it/projects/manage-project-delays.html>
8. KOHT, J., 2010. Sitemaps are the foundation of good information architecture and content strategy [viewed 3 January 2016]. Available from:
<https://www.kohactive.com/blog/building-sitemaps/>
9. MORLEY, M., 2012. What Is Important About Understanding and Continuously Refining a Project's Scope? In: Small Business. 12 July 2012 [viewed 2 January 2016]. Available from:
<http://smallbusiness.chron.com/important-understanding-continuously-refining-projects-scope-30837.html>
10. NOYES, K., 2012. 10 Reasons Open Source Is Good For Business. In: PC World Blog. 21 November 2012 [viewed 4 January 2016]. Available from:
http://www.pcworld.com/article/209891/10_reasons_open_source_is_good_for_business.html
11. OLSEN, E., 2011. Strategic Planning Kit For Dummies. 2nd ed. United Kingdom: John Wiley & Sons; 2nd Edition edition.
12. PORNIN, T., 2011. Why passwords should be hashed. In: IT Security Community Blog. 01 November 2011 [viewed 2 January 2016]. Available from: <http://security.blogoverflow.com/2011/11/why-passwords-should-be-hashed/>
13. PRIORITY., 2013. Priority Management: The Importance of Project Planning [viewed 2 January 2016]. Available from:
http://www.prioritymanagement.com/nsw/resources/resource.php?resource_id=66

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

14. RAMPTON, J., 2014. How Important Is Responsive Design? In: The Business Blog. 5 June 2015 [viewed 3 January 2016]. Available from:
http://www.huffingtonpost.com/john-rampton/business-mobile-responsive-design_b_5267077.html
15. ROUSE, M., 2011. Sprint (software development) definition [viewed 3 January 2016]. Available from:
<http://searchsoftwarequality.techtarget.com/definition/Scrum-sprint>
16. SEAGUE TECHNOLOGIES., 2015. 8 Benefits of Agile Software Development [viewed 3 January 2016]. Available from:
<http://www.seguetech.com/blog/2013/04/12/8-benefits-of-agile-software-development>
17. VIRGINIA COLLEGE., 2006. Blocking Brute Force Attacks [viewed 3 January 2016]. Available from:
https://www.cs.virginia.edu/~csadmin/gen_support/brute_force.php
18. WATERS, K., 2007. 10 Good Reasons To Do Agile Development [viewed 3 January 2016]. Available from: <http://www.allaboutagile.com/10-good-reasons-to-do-agile-development/>

Bibliography

1. CHERRY, K., 2007. Gestalt Laws of Perceptual Organization. *About: Psychology*, 21 March, 1.
2. CRESPO, G, 2013. Responsive Web Design with jQuery. Birmingham: Packt Publishing.
3. FIRDAUS, T, 2013. Responsive Web Design by Example. Birmingham: Packt Publishing.
4. HORIZON INTERACTIVE AWARDS, n.d. A Discussion About Responsive Web Design – Fahrenheit Studio [online] [view 4 January 2015]. Available from: <https://www.horizoninteractiveawards.com/news/article/a-discussion-about-responsive-web-design-fahrenheit-studio>
5. KAYLA KNIGHT, 2011. Responsive Web Design: What It Is and How To Use It. In: Smashing Magazine [online] [2 January 2015]. Available from: <http://www.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/>
6. MAGUIRE, M. and BEVAN, N., 2002. *User requirements analysis* [online]. Montreal, Canada: Kluwer Academic Publishers [viewed 21 August 2015]. Available from: http://www.nigelbevan.com/papers/WCC_UserRequirements.pdf
7. MATT KERSLEY, 2013. Responsive Design Testing [online] [2 January 2015]. Available from: <http://mattkersley.com/responsive/>
8. NORMAN, D., 1991. Cognitive Artifacts. In: Carroll, John M. (ed.). *Designing Interaction: Psychology at the Human-Computer Interface*. Cambridge, UK: Cambridge University Presspp. 17-38.

9. SHELDON, R., 2009. Qualitative or quantitative – which method is for you? [online] [viewed 21 April 2015]. Available from:
<http://www.marketingdonut.co.uk/marketing/market-research/qualitative-or-quantitative-which-method-is-for-you->
10. SUZUNO, M, 2014. What Is Interaction Design? A Guide To The Basics [online] [viewed 21 April 2015]. Available from:
<http://blog.aftercollege.com/2014/interaction-design-guide-basics/>
11. TOM MASLEN, 2014. Prioritizing Devices: Testing And Responsive Web Design. In: Smashing Magazine [online] [2 January 2015]. Available from:
<http://www.smashingmagazine.com/2014/07/14/testing-and-responsive-web-design/>
12. TRIPP, A., 2014. Color Theory: The Importance Of Color In Web Design. In: *Design & Promote* [online]. 27 June 2014 [viewed 23 April 2015]. Available from: <http://www.designandpromote.com/color-theory-the-importance-of-color-in-web-design/>
13. U.S. DEPARTMENT OF HEALTH & HUMAN SERVICES, 2001. Usability Evaluation Basics [online] [viewed 21 April 2015]. Available from:
<http://www.usability.gov/what-and-why/usability-evaluation.html>
14. WARNER, J and D.LAFONTAIN, 2010. Mobile Web Design for Dummies. United Kingdom: For Dummies.

Appendix F: User Reference Guide

User Reference Guide

This guide is to be used in the event that a client has chosen to install the Hubbal staff attendance tracking/organisation software to help prioritise their staff's involvement with attendance and scheduling and how they can interact between one another, all under one roof or application.

The table below is to be used to determine the amount of users that the application should be limited to based on the technological capabilities of the server that your organisation has installed the application on.

RAM In Gigabytes	Processor Manufacturer	Bits	Core	User Cap
1	AMD Sempron	64-bit	Single	100
1	Intel Celeron	32-bit/64-bit	Single	150
2	AMD Athlon 64	32-bit/64-bit	Single	200
2	Intel Pentium 4	64-bit	Single	200
3	AMD Athlon 64 X2	32-bit	Dual	225
3	Intel Pentium D	64-bit	Dual	225
4	AMD Sempron	64-bit	Single	150
4	Intel Celeron	32-bit/64-bit	Single	200
5	AMD Athlon 64	32-bit/64-bit	Single	250
5	Intel Pentium 4	64-bit	Single	300

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

6	AMD Athlon 64 X2	32-bit	Dual	300
6	Intel Pentium D	64-bit	Dual	300
7	AMD Sempron	64-bit	Single	400
7	Intel Celeron	32-bit/64-bit	Single	425
8	AMD Athlon 64	32-bit/64-bit	Single	450
8	Intel Pentium 4	64-bit	Single	500

These are simple estimates made and although may soon be out of date, at the production of the application, these were estimated setups that the application could be installed on for an organisation. Most common setups should have the ability to allow the application to function.

These numbers are to be used if your organisation is absent of those who do not understand computing limitations, if you have an IT team or computing faculty available, they can use these numbers to determine if the application is suitable or not for installation.

Appendix G: Project Definition Specification

Specification

A specification is used to fully outline the expectations of the project from start to finish. Without a specification, there would be nothing to evaluate the level of success of the final project outcome.

Description

A attendance tracking system with the ability to monitor and backtrack attendance records with grouping support, included will be a file management system that can structure a course/class of the educational establishment that the system is installed for.

Features

The primary function of the final creation will help businesses and organisations track their attendance records through the use of registration areas, supported by rankings and groupings to differentiate users from each other.

The system will also offer support for a file storage unit for students who have unattended a particular session, this can also benefit catch-up capabilities for students who may have not attended a class or would like revision resources to be readily available online.

External Interface Requirements

i. User Interface

- a. Follow conventions of standard Dashboard styles & templates.
- b. Left hand layout with the ability to be hidden.
- c. Responsive design to support multiple devices.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

- d. Neutral colour design to suit all types of users.
- e. Typical horizontal navigation to contain User-specific information.

ii. Hardware

- a. Moderately powered device to contain Ruby on Rails server, 2GB RAM and memory of at least 4GB.
- b. Screen needed to control server and make changes of front-end if needed.
- c. Mouse and keyboard are also needed.
- d. Networking abilities to connect to the internet.

iii. Software

- a. Administration Privileges to control installation of server.
- b. Web Development Software to make front-end changes if necessary.
- c. Ability to run background processes to support Ruby on Rails server.

Other Non-functional Requirements

i. Performance

- a. The speed of the system is based entirely on the device that the software is performing on. This is why a baseline has been specified above in the Hardware requirements.
- b. Operating system has no effect on performance, providing that there is a sufficient device for the software to function on.

ii. Safety

- a. Health and safety terms are based purely on the policies written by the business using the product.
- b. Usual safety precautions must be taken such as Data Protection and filter for curse or unacceptable language.

iii. Security

- a. Data Protection Act Compliance.
- b. Computer Misuse Act Compliance.
- c. Copyright Designs and Patents Act Compliance.

Appendix H: Agile Methodology Cheat Sheet

Agile Project Management For Dummies

From [Agile Project Management For Dummies](#) by Mark C. Layton



Agile project management focuses on continuous improvement, scope flexibility, team input, and delivering essential quality products. Agile project management methodologies include scrum, extreme programming (XP), and lean, among others. These methodologies all adhere to the Agile Manifesto and the 12 Agile Principles, which focus on people, communications, the product, and flexibility.

A Manifesto for Agile Software Developers

The Agile Software Development Manifesto© is an intentionally streamlined expression of the core values of agile project management. Use this manifesto as a guide to implement agile methodologies in your projects.

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work, we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more."

©Agile Manifesto Copyright 2001: Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas.

This declaration may be freely copied in any form, but only in its entirety through this notice.

The 12 Agile Principles

The 12 Agile Principles are a set of guiding concepts that support project teams in implementing agile projects. Use these concepts to implement agile methodologies in your projects.

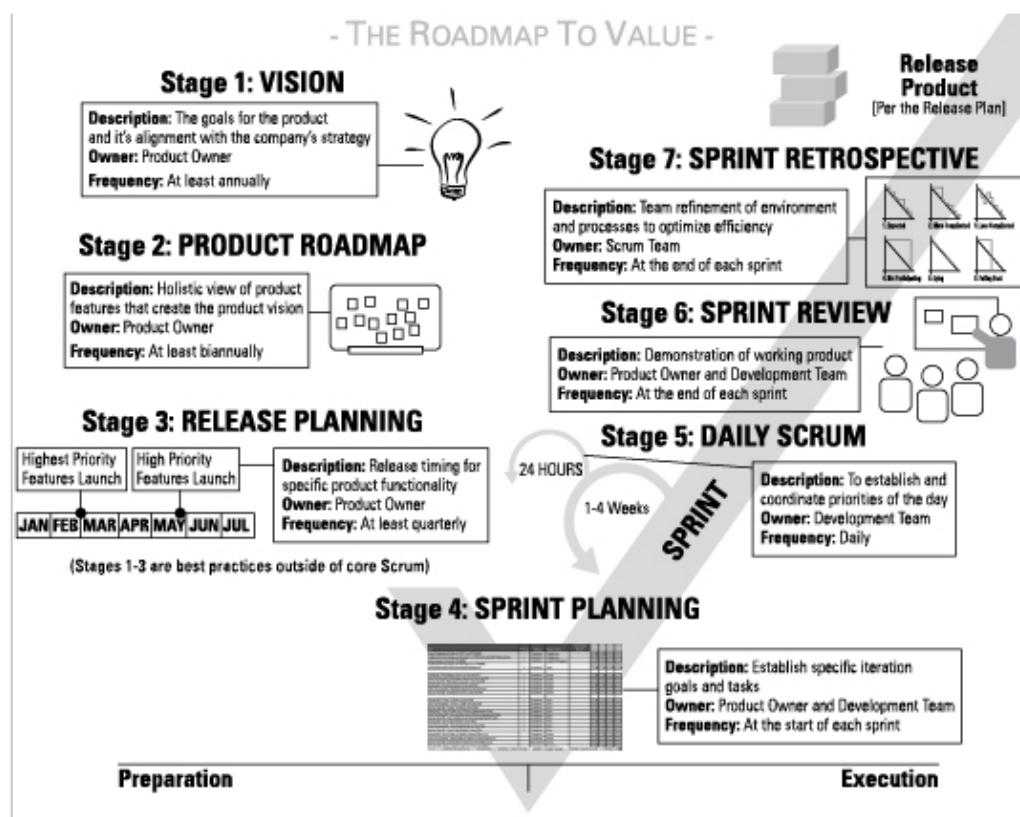
- 1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.**
- 2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.**
- 3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.**
- 4. Business people and developers must work together daily throughout the project.**
- 5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.**
- 6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.**
- 7. Working software is the primary measure of progress.**
- 8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.**
- 9. Continuous attention to technical excellence and good design enhances agility.**
- 10. Simplicity — the art of maximizing the amount of work not done — is essential.**
- 11. The best architectures, requirements and designs emerge from self-organizing teams.**

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

The Agile Roadmap to Value

The Roadmap to Value is a high-level view of an agile project. The stages of the Roadmap to Value are described in the list following the diagram:



- In Stage 1, the product owner identifies the *product vision*. The product vision is a definition of what your product is, how it will support your company or organization's strategy, and who will use the product. On longer projects, revisit the product vision at least once a year.
- In Stage 2, the product owner creates a *product roadmap*. The product roadmap is a high-level view of the product requirements, with a loose time

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

A **product roadmap** is a high-level view of the product requirements, with a loose time frame for when you will develop those requirements. Identifying product requirements and then prioritizing and roughly estimating the effort for those requirements are a large part of creating your product roadmap. On longer projects, revise the product roadmap at least twice a year.

- In Stage 3, the product owner creates a *release plan*. The release plan identifies a high-level timetable for the release of working software. An agile project will have many releases, with the highest-priority features launching first. A typical release includes three-to-five sprints. Create a release plan at the beginning of each release.
- In Stage 4, the product owner, the master, and the development team plan sprints, also called iterations, and start creating the product within those sprints. *Sprint planning* sessions take place at the start of each sprint, where the scrum team determines what requirements will be in the upcoming iteration.
- In Stage 5, during each sprint, the development team has *daily meetings*. In the daily meeting, you spend no more than 15 minutes and discuss what you completed yesterday, what you will work on today, and any roadblocks you have.
- In Stage 6, the team holds a *sprint review*. In the sprint review, at the end of every sprint, you demonstrate the working product created during the sprint to the product stakeholders.
- In Stage 7, the team holds a *sprint retrospective*. The sprint retrospective is a meeting where the team discusses how the sprint went and plans for improvements in the next sprint. Like the sprint review, you have a sprint retrospective at the end of every sprint.

Agile Project Management Roles

It takes a cooperative team of employees to complete a project. Agile project teams are made up of many people and include the following five roles:

- **Development team:** The group of people who do the work of creating a

product. Programmers, testers, designers, writers, and anyone else who has a hands-on role in product development is a member of the development team.

- **Product owner:** The person responsible for bridging the gap between the customer, business stakeholders, and the development team. The product owner is an expert on the product and the customer's needs and priorities. The product owner works with the development team daily to help clarify requirements. The product owner is sometimes called a *customer representative*.
- **Scrum master:** The person responsible for supporting the development team, clearing organizational roadblocks, and keeping the agile process consistent. A scrum master is sometimes called a *project facilitator*.
- **Stakeholders:** Anyone with an interest in the project. Stakeholders are not ultimately responsible for the product, but they provide input and are affected by the project's outcome. The group of stakeholders is diverse and can include people from different departments, or even different companies.
- **Agile mentor:** Someone who has experience implementing agile projects and can share that experience with a project team. The agile mentor can provide valuable feedback and advice to new project teams and to project teams that want to perform at a higher level.

Agile Project Management Artifacts

Project progress needs to be measurable. Agile project teams often use six main artifacts, or deliverables, to develop products and track progress, as listed here:

- **Product vision statement:** An elevator pitch, or a quick summary, to communicate how your product supports the company's or organization's strategies. The vision statement must articulate the goals for the product.
- **Product backlog:** The full list of what is in the scope for your project, ordered by priority. Once you have your first requirement, you have a *product backlog*.

product backlog.

- **Product roadmap:** The product roadmap is a high-level view of the product requirements, with a loose time frame for when you will develop those requirements.
- **Release plan:** A high-level timetable for the release of working software.
- **Sprint backlog:** The goal, user stories, and tasks associated with the current sprint.
- **Increment:** The working product functionality at the end of each sprint.

Agile Project Management Events

Most projects have stages. Agile projects include seven events for product development. These events are meetings and stages and are described in the following list:

- **Project planning:** The initial planning for your project. Project planning includes creating a product vision statement and a product roadmap, and can take place in as little time as one day.
- **Release planning:** Planning the next set of product features to release and identifying an imminent product launch date around which the team can mobilize. On agile projects, you plan one release at a time.
- **Sprint:** A short cycle of development, in which the team creates potentially shippable product functionality. Sprints, sometimes called *iterations*, typically last between one and four weeks. Sprints can last as little as one day, but should not be longer than four weeks. Sprints should remain the same length throughout the entire projects.
- **Sprint planning:** A meeting at the beginning of each sprint where the scrum team commits to a sprint goal. They also identify the requirements that support this goal and will be part of the sprint, and the individual tasks it will take to complete each requirement.
- **Daily scrum:** A 15-minute meeting held each day in a sprint, where development team members state what they completed the day before,

what they will complete on the current day, and whether they have any roadblocks.

- **Sprint review:** A meeting at the end of each sprint, introduced by the product owner, where the development team demonstrates the working product functionality it completed during the sprint.
- **Sprint retrospective:** A meeting at the end of each sprint where the scrum team discusses what went well, what could change, and how to make any changes.

Agile Project Management Organizations, Certifications, and Resources

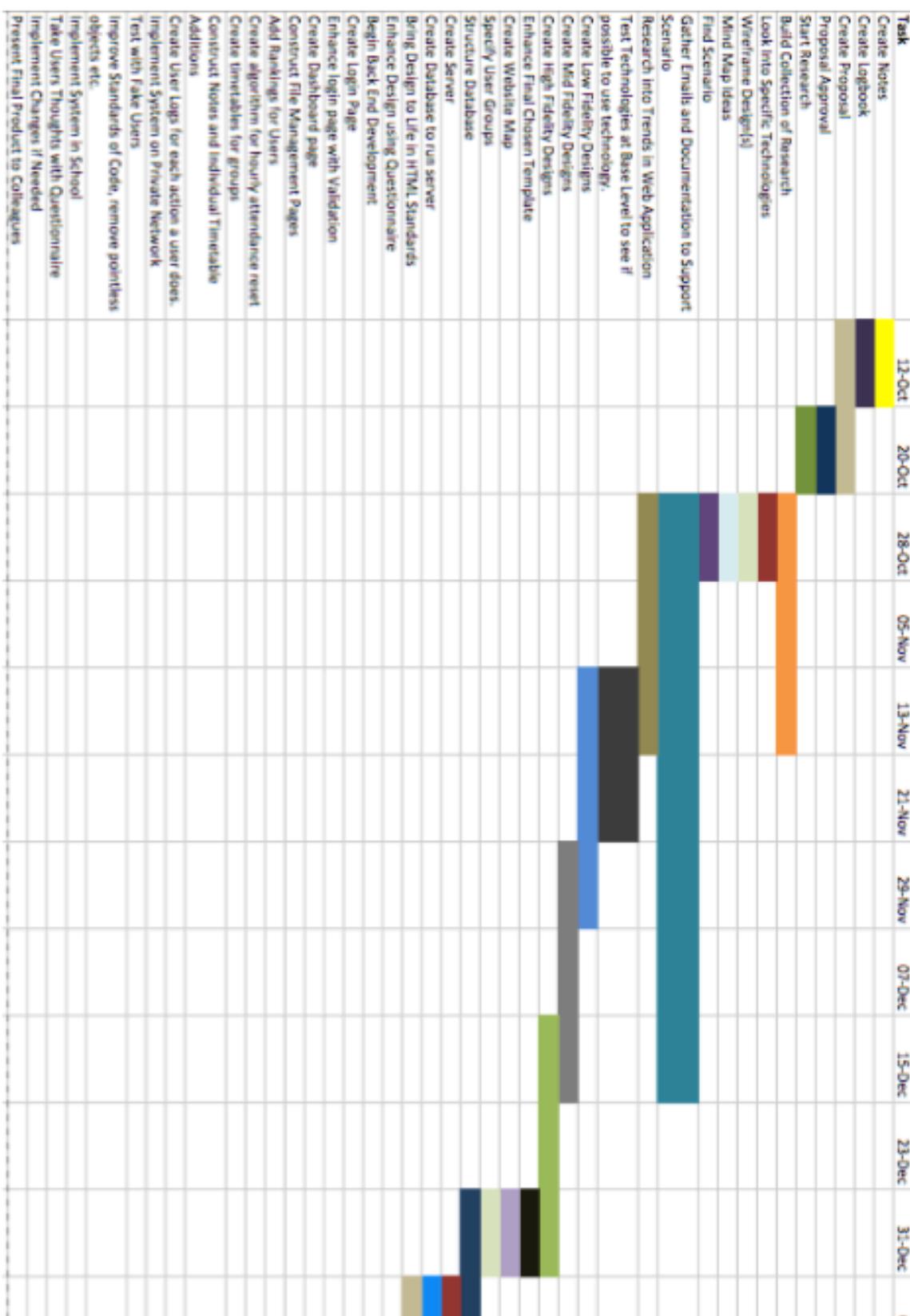
There is a big agile project management world out there. Here are a few of the useful links to members of the agile practitioner community:

- **Agile Alliance:** The Agile Alliance is the original global agile community, with a mission to help advance agile principles and practices, regardless of methodology.
- **Scrum Alliance:** The Scrum Alliance is a nonprofit professional membership organization that promotes understanding and usage of scrum. The Scrum Alliance offers a number of professional certifications:
 - Certified Scrum Master (CSM)
 - Certified Scrum Product Owner (CSPO)
 - Certified Scrum Developer (CSD)
 - Certified Scrum Professional (CSP)
 - Certified Scrum Coach (CSC)
 - Certified Scrum Trainer (CST)
- **XProgramming.com:** Ron Jeffries, one of the originators of the extreme programming (XP) development approach, provides resources and services in support of XP's advancement on the XProgramming.com site.
- **Lean Essays:** Lean Essays is a blog from Marv and Tom Ponnendieck.

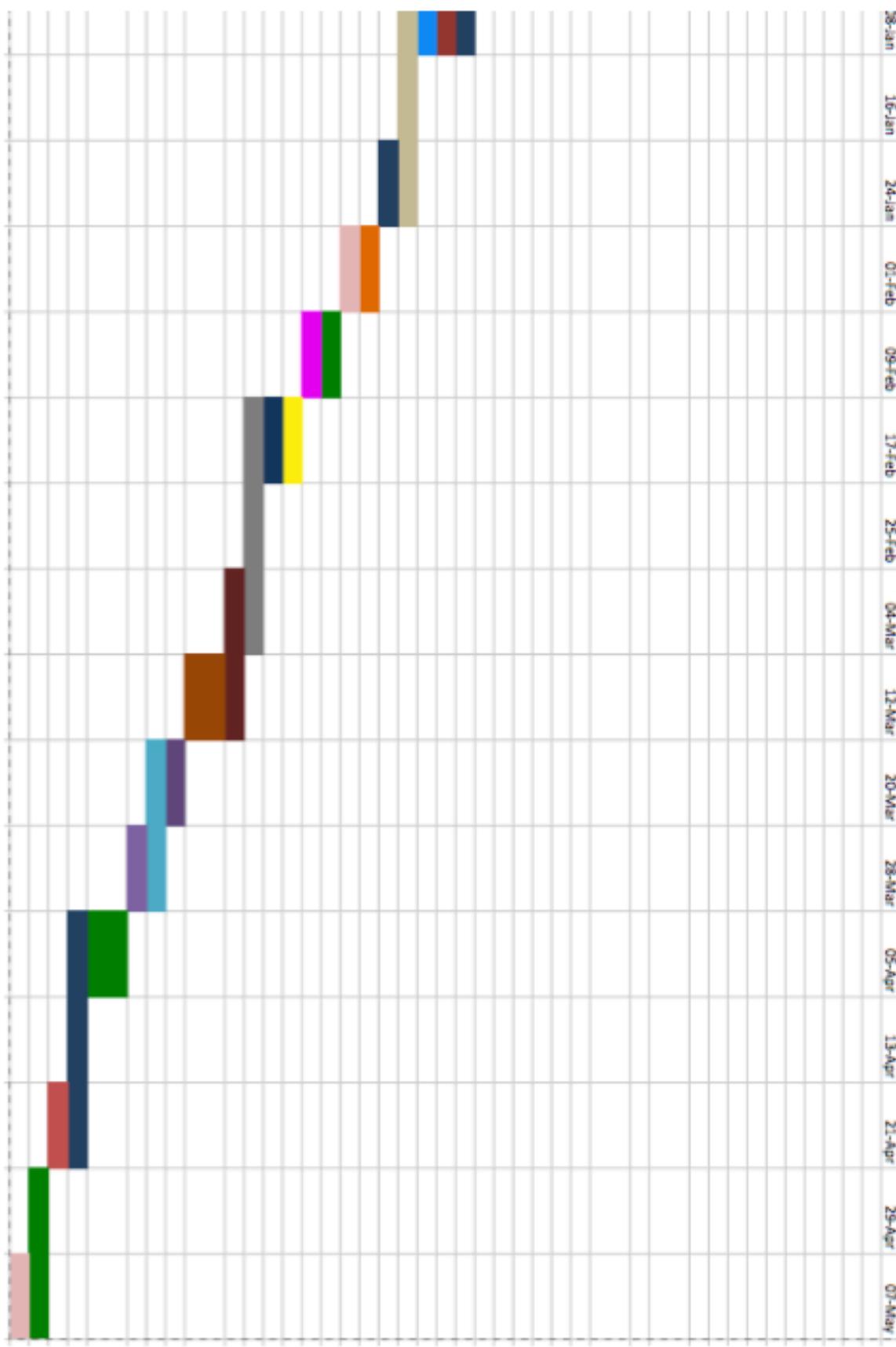
CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

- [Lean Essays](#) - [Lean Essays](#) is a blog community and forum opportunity for thought leaders in the use of lean concepts within the software development space.
- **PMI Agile Community:** The Project Management Institute (PMI) is the largest nonprofit project management membership association in the world. The agile section of PMI's website provides access to papers, books, and seminars about agile project management. PMI supports an agile community of practice and a certification, the PMI Agile Certified Practitioner (PMI-ACP).
- **Platinum Edge:** Since 2001, my team at Platinum Edge has been helping companies successfully take their project management practices to a higher level. We provide training classes worldwide and also develop transition strategies and coaching for organizations moving to agile project management. Visit the training section of our site to find an upcoming Certified Scrum Master, Certified Scrum Product Owner, PMI-ACP preparation, or agile overview class near you.

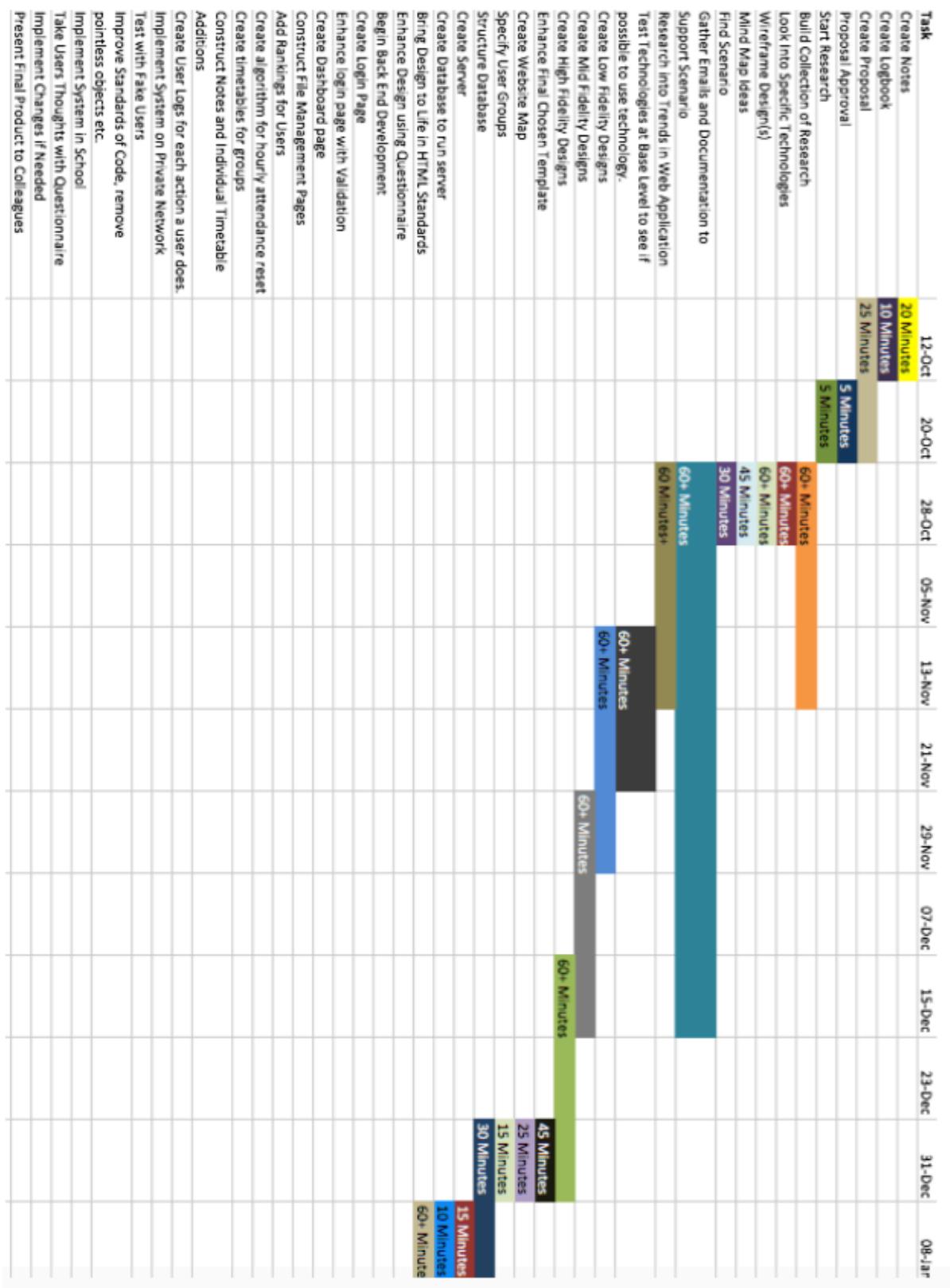
Appendix I: Early Gantt Chart



CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?



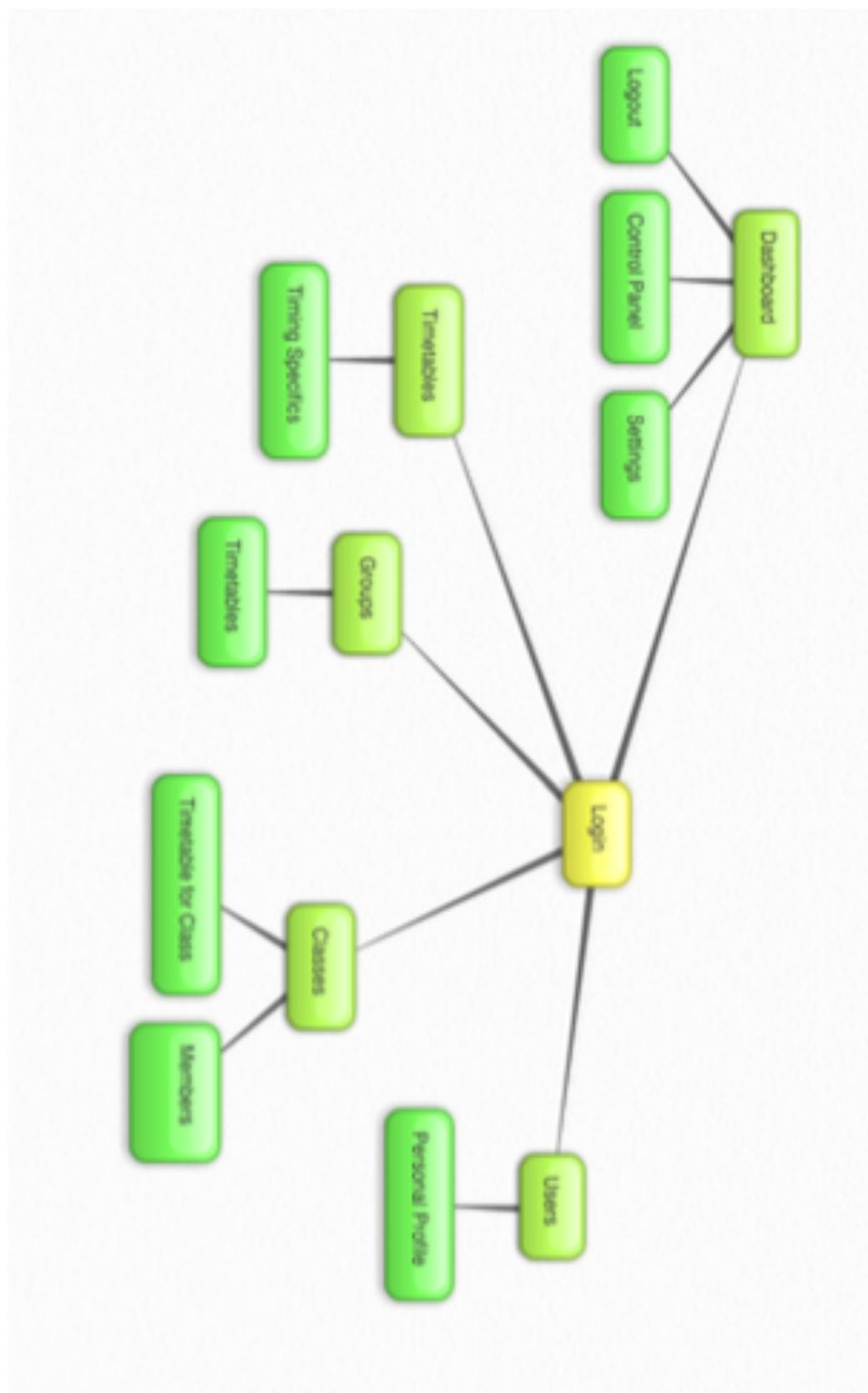
Appendix J: Project Review Gantt Chart



CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?



Appendix K: Mind Map



CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Appendix L: Design Fidelity Grid

Fidelity	Appearance	Optimal uses	Advantages	Limitations
Low	Rough sketch; highly schematic and approximate. Little or no interactive functionality.	Early design; conceptualizing and envisioning the application.	Low cost; useful communication vehicle; proof of concept.	Limited usefulness after requirements established; limitations in usability testing
Mid	Fairly detailed and complete but objects are presented in schematic or approximate form. Provides simulated interactive functionality and full navigation.	Designing and evaluating most interactive aspects, including navigation, functionality, content, layout and terminology.	Much lower cost and time as compared to high fidelity; detail is sufficient for usability testing; serves as a reference for the functional specification.	Does not fully communicate the look and feel of the final product; some limitations as a specification document.
High	Lifelike simulation of the final product; refined graphic design. Highly functional, but the back end might be simulated rather than real.	Marketing tool; training tool; simulation of advanced or highly interactive techniques.	High degree of functionality; fully interactive; defines look and feel of final product; serves as a living specification.	Expensive to develop; time consuming to build.

Appendix M: Backend Teacher Persona



Josie Simpson Geography Teacher

Josie is a full-time Geography teacher holding a degree in Travel & Tourism, her past experiences include a year around Asia during a gap year from University to enhance her knowledge and understanding of different cultures around the world and feels she can convey her earned knowledge to children.

Interests

- Travel
- Learning
- Swimming
- Fitness

Goals & Needs

Josie desires to become a University lecturer to teach more mature students about the benefits of travelling and how different cultures around the world can teach you about the development of the planet.

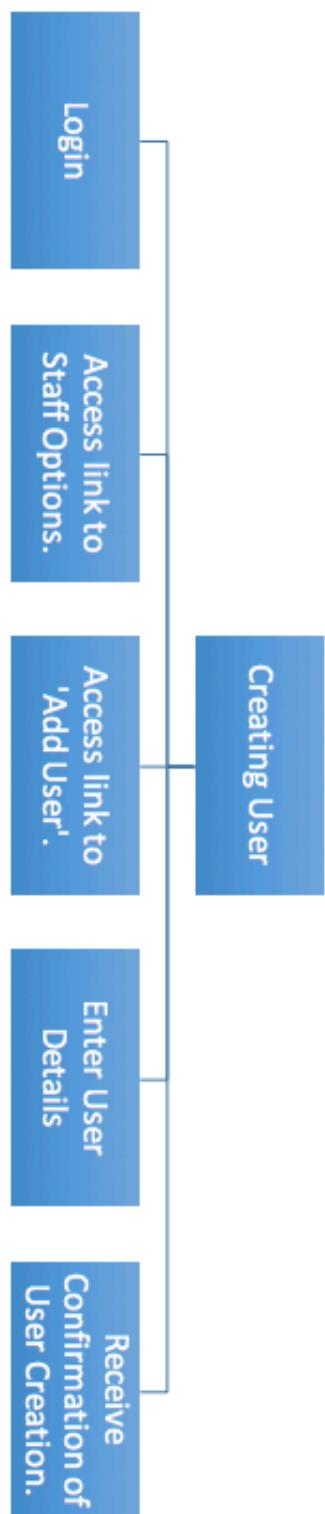
Scenario

Josie has multiple classes and needs a way or organising her schedule. She logs into Hubbal and navigates to the Timetable section to input her weekly schedule into her profile.

Appendix N: Backend Design Hierarchical Task Analysis'

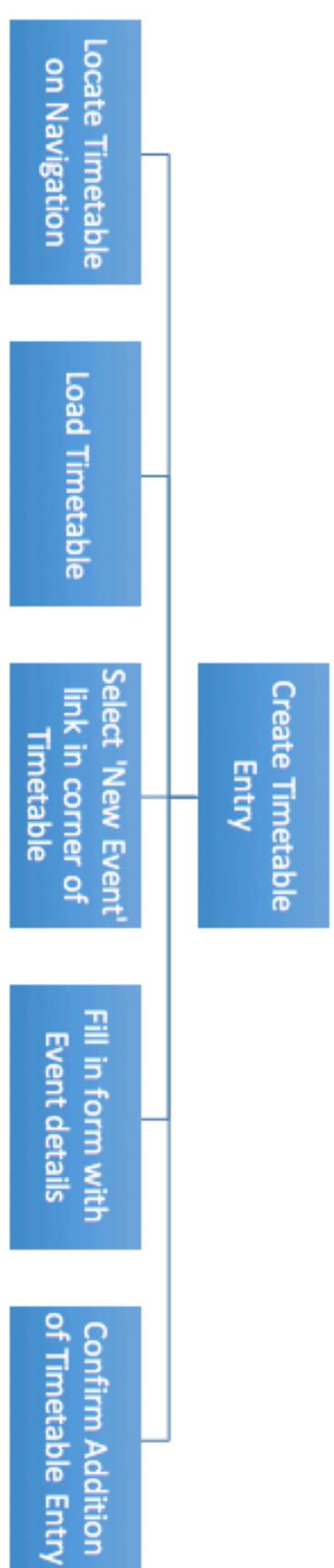
Hierarchical Task Analysis for Creating a New User

This task analysis has been created to visually demonstrate the user journey that needs to be followed for a user to successfully create a new user in the system.



Hierarchical Task Analysis of Creating a Timetable Entry

This task analysis has been created to route the completion of the task that a user can complete of creating a timetable entry, this is a personal task analysis to every user as a timetable belongs to a user and cannot be edited or destroyed.



Hierarchical Task Analysis of Logging into Hubbal

This task analysis has been created to display the intended steps the user must take to successfully login to the Hubbal web application, used to organise staff's lives.

Logging into Hubbal

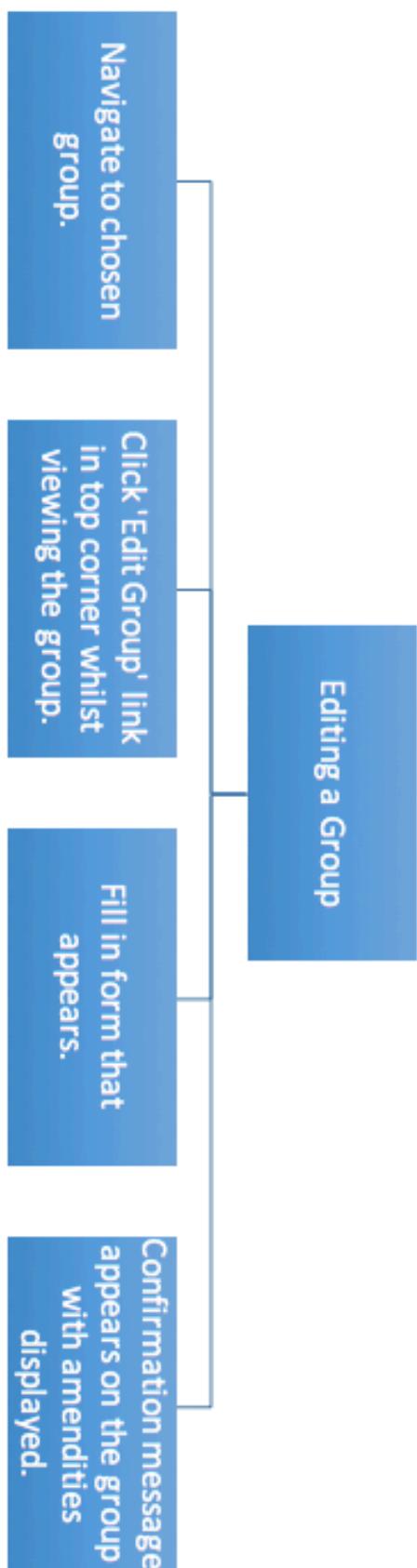
Navigate to '/login' of the hosting URL of the organisation website.

Fill in Login form, e-mail and password.

Confirm login by accessing Dashboard.

Hierarchical Task Analysis of Editing a Group

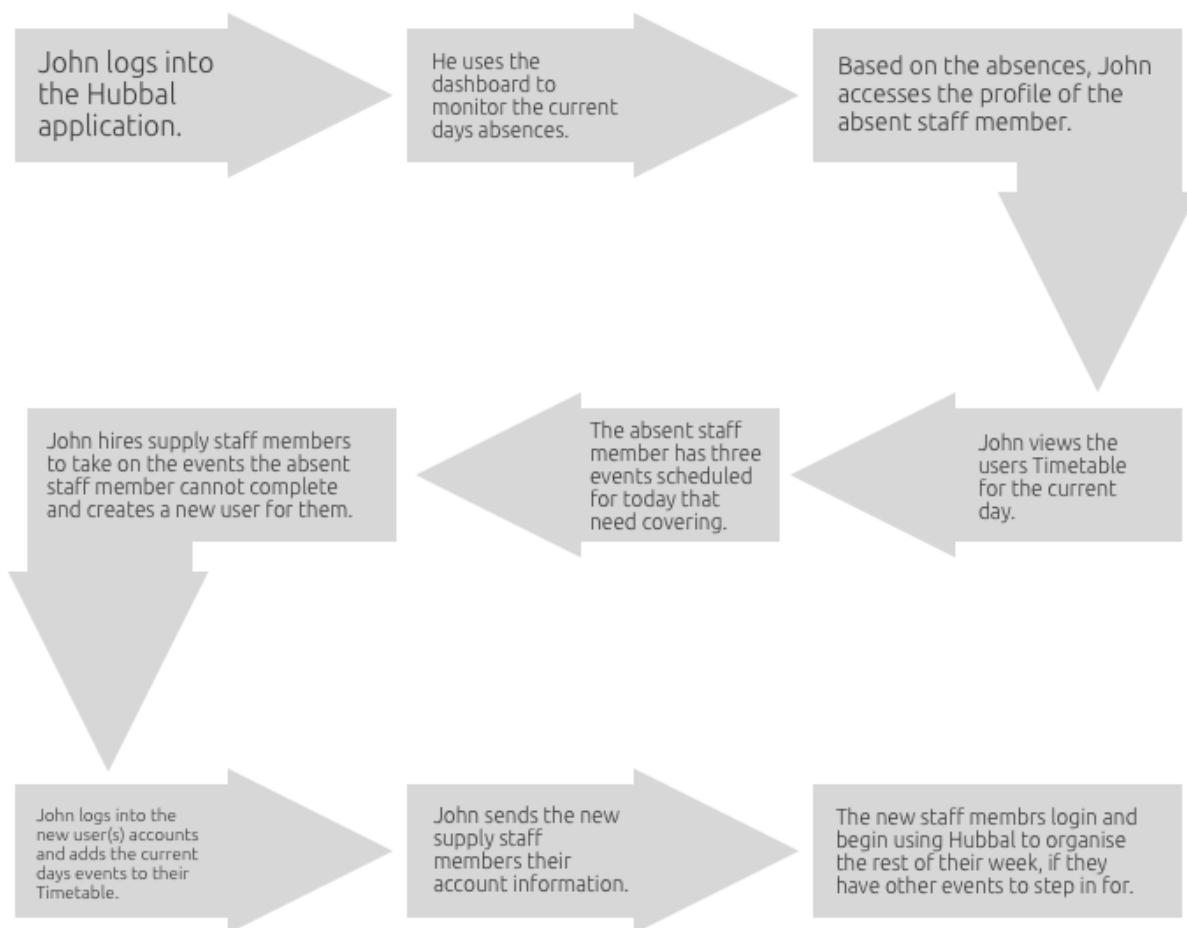
This task analysis follows a user editing a group within Hubbal, although the same task route can be followed to edit any item within the application, as the application will be designed this way to make the application easier to understand for the user.



Appendix O: Backend Design User Journey

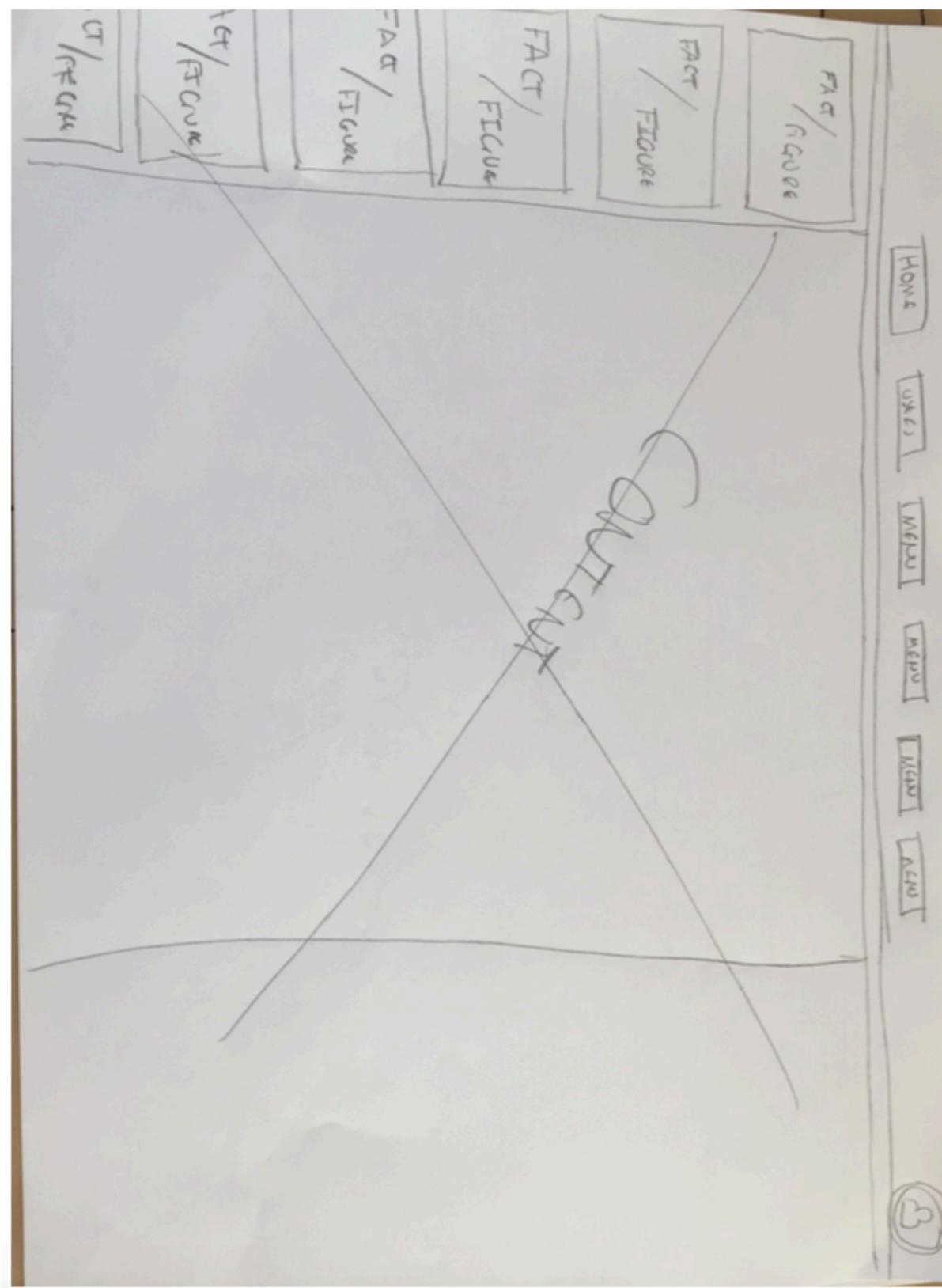
User Journey Backend Design

The User Journey illustrating the intended route the user needs to take to login, view their dashboard and use the design to the intended purpose, these are random events and are simply used to understand the direction the application should be going in with design.

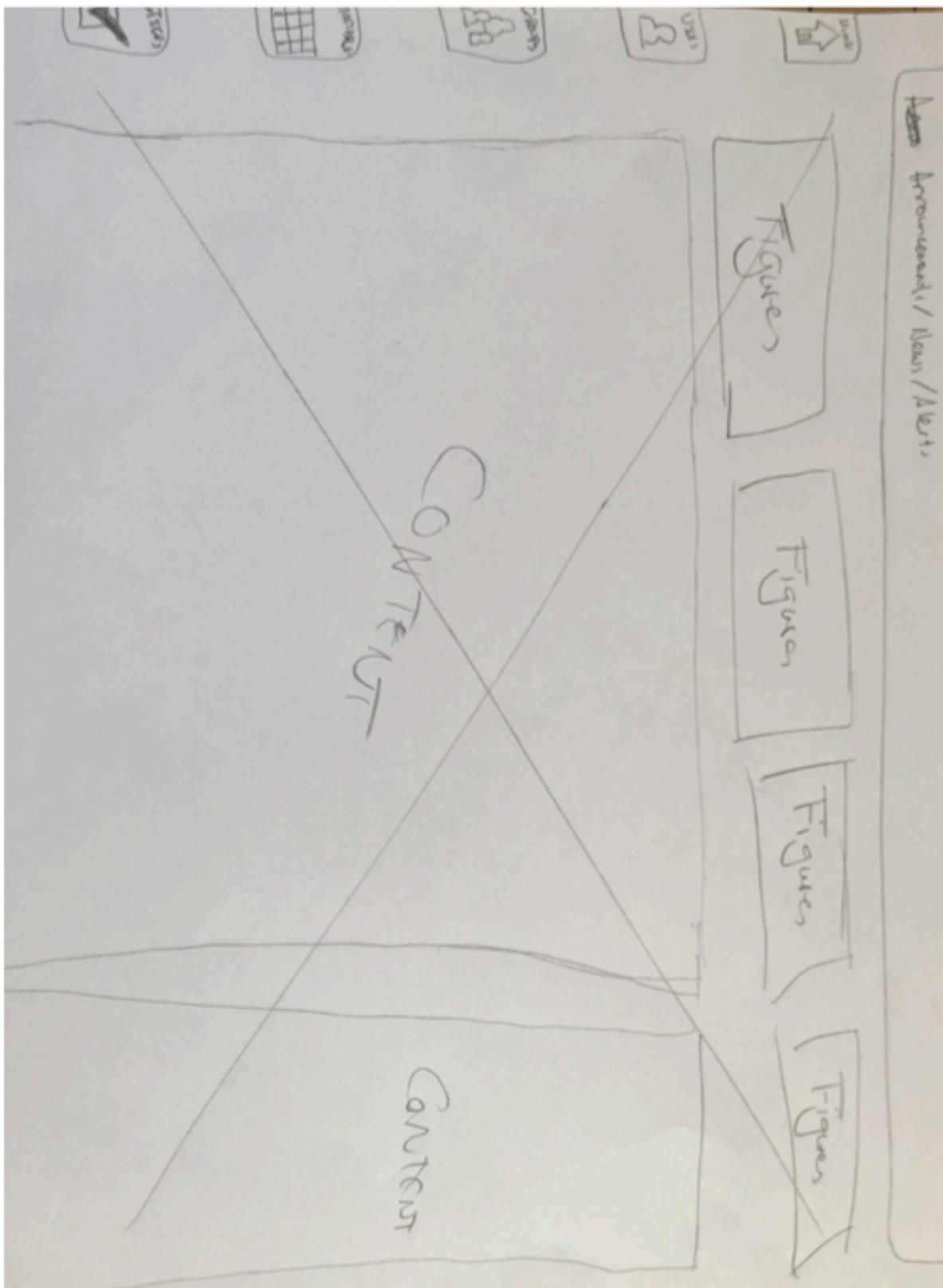


Appendix P: Low Fidelity Backend Designs

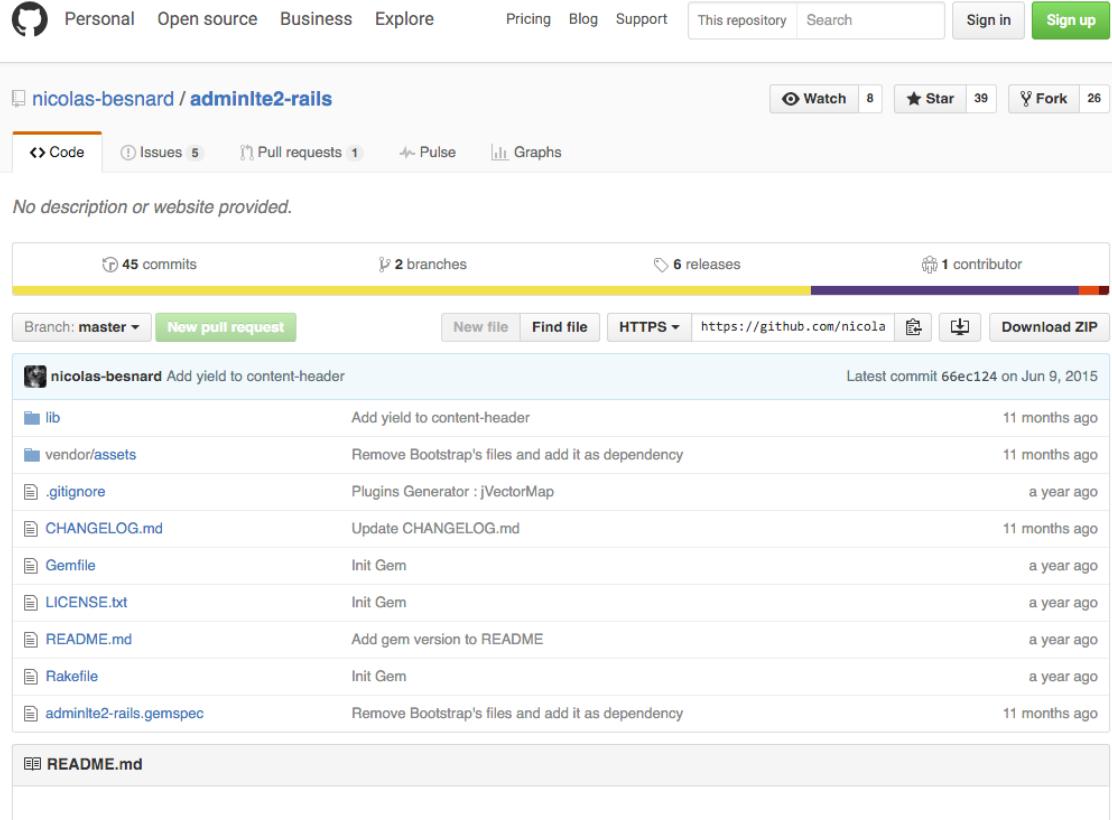
Design One



Design Two



Appendix Q: Admin LTE2 Github Repository



The screenshot shows the GitHub repository page for `nicolas-besnard / adminlte2-rails`. The repository has 45 commits, 2 branches, 6 releases, and 1 contributor. The latest commit was on June 9, 2015. The repository contains files like `lib`, `vendor/assets`, `.gitignore`, `CHANGELOG.md`, `Gemfile`, `LICENSE.txt`, `README.md`, and `adminlte2-rails.gemspec`. The `README.md` file indicates the gem version is v0.0.6.

AdminLTE 2 Rails Gem

AdminLTE is a premium Bootstrap theme for backend.

Installation

Add this line to your application's Gemfile:

```
gem 'adminlte2-rails'
```

And then execute:

```
$ bundle
```

Or install it yourself as:

```
$ gem install adminlte2-rails
```

Then, generate the default template:

```
$ bin/rails generate admin_lte2
```

Plugins

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Add a new plugin

By default, there's no plugin installed. However, I've created a generator to help you install them

```
$ rails g admin_lte2_plugins PLUGIN_NAME
```

Plugins available

- [jVectorMap](#) (`j_vector_map`)
- [ChartJS](#) (`chart_js`)
- [Sparkline](#) (`sparkline`)
- [Bootstrap Slider](#) (`bootstrap_slider`)
- [Bootstrap WYSIHTML5](#) (`bootstrap_wysihtml5`)
- [Full Calendar](#) (`fullcalendar`)
- [Knob](#) (`knob`)
- [TimePicker](#) (`timepicker`)
- [Pace](#) (`pace`)
- [MorrisJS](#) (`morris`)
- [Date Range Picker](#) (`daterangepicker`)
- [Color Picker](#) (`colorpicker`)
- [Date Picker](#) (`datepicker`)
- [slimScroll](#) (`slimScroll`)
- [iCheck](#) (`icheck`)
- [Toastr](#) (`toastr`)

Plugins missing

- ckeditor
- datatables
- fastclick
- flot
- input-mask
- ionslider

Contributing

1. Fork it (<https://github.com/nicolas-besnard/adminlte2-rails/fork>)
2. Create your feature branch (`git checkout -b my-new-feature`)
3. Commit your changes (`git commit -am 'Add some feature'`)
4. Push to the branch (`git push origin my-new-feature`)
5. Create a new Pull Request

Appendix R: High Fidelity Backend Design Views

Login (/login)

The image shows a high-fidelity design of a web-based login interface. At the top center is the Hubbal logo, which consists of a stylized 'H' icon followed by the word 'Hubbal' in a bold, sans-serif font. Below the logo is a white rectangular input field with a thin black border. Inside the field, the placeholder text 'This area is for staff only. Please enter your username and password to begin.' is displayed in a small, gray, sans-serif font. Below this field are two horizontal input fields. The first input field is yellow and contains the text '7TAlderson2' in a black, sans-serif font. To its right is a small blue icon of a person's head with a video camera over it. The second input field is also yellow and contains the text '.....' (represented by five dots) in a black, sans-serif font. To its right is a small blue icon of a computer monitor with a lock symbol. Below these two input fields is a solid blue rectangular button with the text 'Log in' written in a white, sans-serif font.

Dashboard (/dashboard)

The screenshot displays a dashboard interface with the following sections:

- User Profile:** Tom Alderson, Last Login: 23/04/2016 - 11:23.
- Navigation:** Main Website, Dashboard, Users, Groups (1), Timetable, Courses/Classes.
- Dashboard Summary:** Everything important all in one place.
- Statistics:**
 - USERS:** 1/500 (0% of maximum users)
 - COURSES/CLASSES:** 1/200 (1% of maximum courses/classes)
 - LOGGED IN:** 1 (1 user)
- Today's Agenda:**

Time	Task
08:00	Website Development
09:00	Website Development
- Last 10 Logins:**

Name	Date & Time
Tom Alderson	23/04/2016 - 11:25
Tom Alderson	22/04/2016 - 19:56
Tom Alderson	10/04/2016 - 11:25
Tom Alderson	08/04/2016 - 15:27
Tom Alderson	08/04/2016 - 14:46
- Your Notes:**

Title	Preview
Shopping List	Eggs, Milk, Bread
Website Designs	A couple of design ideas th...
Title...	
$\star \quad B \quad I \quad U \quad \mathfrak{S} \quad x_2 \quad x^2 \quad A \curvearrowleft T \curvearrowright \quad \exists \curvearrowleft \exists \curvearrowright \quad \equiv \curvearrowleft \equiv \curvearrowright \quad \gg \curvearrowleft \quad \sim$	

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

LOGINS TODAY
1

TOTAL NOTES
9

1 users logged in out of 1 users.

Reason for Absence

Absent Until

Created

23/04/2016

Date & Time

April 23, 2016

0:27pm

① - Calendar

Su	Mo	Tu	We	Th	Fr	Sa
1	2					
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Messages Pages

Inbox (/messages)

The screenshot shows the 'Inbox' page of the Hubbal web application. At the top, there's a header bar with the 'Hubbal' logo, a user profile for 'Tom Alderson' (Last Login: 23/04/2016 - 11:25), and a navigation menu with links like 'Main Website', 'Dashboard', 'Messages', 'Users', 'Groups (1)', 'Timetable', and 'Courses/Classes'. Below the header is a sidebar titled 'NAVIGATION' containing the same links. The main content area is titled 'Inbox' with the subtitle 'All of your received messages.' It includes columns for 'Sent By', 'Subject', and 'Message Preview'. A message preview is shown: 'You currently have no messages in your inbox.' On the far left, there's a vertical sidebar with the text 'Copyright © 2016 Hubbal. All rights reserved.'

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

The screenshot shows a web-based application interface. At the top right, there is a user profile for "Tom Alderson" with a small profile picture, a "Logout" button, and a gear icon for settings. Below the header, a large blue sidebar on the right contains the "Hubbal" logo, the user's name "Tom Alderson", and a "Last Login: 28/04/2016 - 11:21". The sidebar also features a "New Message" button with a plus sign and a gear icon. To the left of the sidebar, a vertical navigation menu lists "Main Website", "Dashboard", "Messages", "Users", "Groups (1)", "Timetable", and "Courses/Classes". The main content area is titled "Content" and displays a message: "Your inbox." The bottom left corner of the main content area has a small note: "Developed by Thomas Alderson".

Outbox (/message/outbox)

The screenshot shows the "Outbox" section of the application. The title "Outbox" is at the top, followed by the subtitle "Messages you have sent". On the left, there is a "Sent To" field with a placeholder "Enter recipient email address". Below this, a list of recipients is shown with their names and profile pictures: "Tom Alderson" (last login 28/04/2016), "Main Website", "Dashboard", "Messages", "Users", "Groups (1)", "Timetable", and "Courses/Classes". The bottom left corner of the main content area has a small note: "Copyright © 2016 Hubbal.."

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

All rights reserved.

e sent to other users.

Subject

You currently have no messages in your outbox.

Content



Devi

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Developed by Thomas Alderson



Tom Alderson



Users (/users)

The screenshot shows the Hubbal web application interface. The top navigation bar is blue and contains the Hubbal logo (a stylized 'H' icon), the word 'Hubbal', and a three-dot menu icon. On the right side of the header, there is a user profile section for 'Tom Alderson' (Last Login: 23/04/2016 - 11:25). Below the header is a dark sidebar with a navigation menu:

- Main Website
- Dashboard
- Messages
- Users
- Groups (1)
- Timetable
- Courses/Classes

The main content area is titled 'Users' and features a sub-header: 'A list of every single user within the system.' It displays a card for 'Tom Alderson' (7TAlderson2, Administrator) with a circular profile picture.

Copyright © 2016 Hubbal. All rights reserved.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Developed by Thomas Alderson



Tom Alderson



Group Page (/groups/*)

The screenshot displays a web application interface for 'Hubbal'. The top navigation bar includes a logo, the word 'Hubbal', and a user profile for 'Tom Alderson' with a last login timestamp. A sidebar on the left lists navigation links: Main Website, Dashboard, Messages, Users, Groups (1), Timetable, and Courses/Classes. The main content area shows a summary for a group managed by Tom Alderson, detailing basic information like Name (Administrator) and Leader (Tom Alderson), and showing 1 total member. A section titled 'People in this Group' lists Tom Alderson.

Basic Information

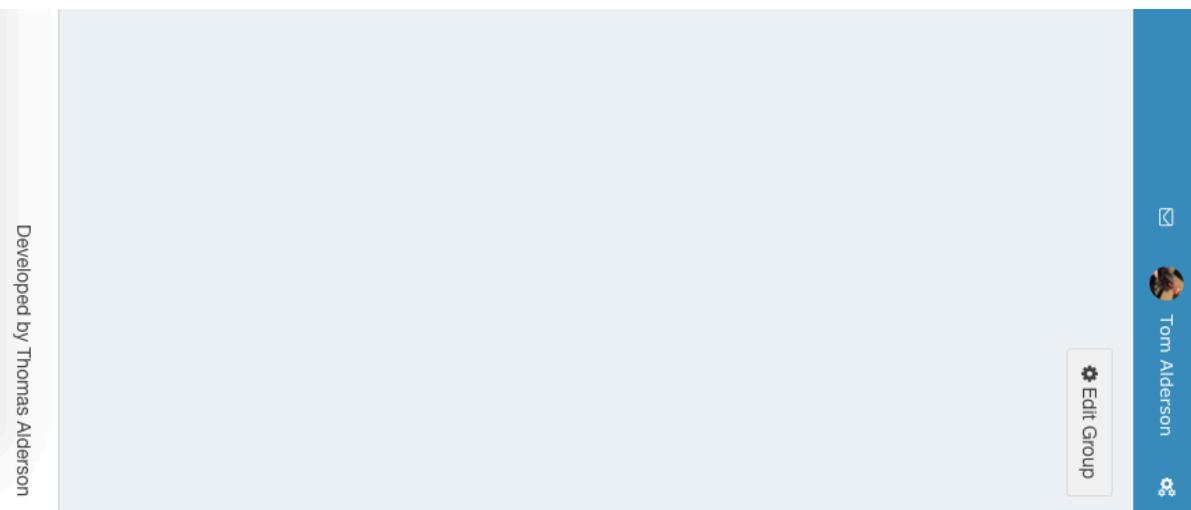
Name	Administrator
Leader	Tom Alderson
Total Members	1

People in this Group

Tom Alderson

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Developed by Thomas Alderson



CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Timetable (/timetables/*)

The screenshot shows the Hubbal web interface with a dark blue header and sidebar. The header displays the title 'CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?' and the date '23/04/2016 - 11:23'. The sidebar on the left includes a user profile for 'Tom Alderson' (Last Login: 23/04/2016 - 11:23), navigation links for 'Main Website', 'Dashboard', 'Messages', and 'Users', and a 'Groups (1)' section. The main content area is titled 'Timetable' with the subtitle 'Your personal schedule.' Below this are three buttons: '- Statistics', 'Hours Today', and 'Weekly Hours' (which is highlighted with a blue border). The central part of the screen shows a 'Timetable' grid from 08:00 to 17:00 on Monday through Thursday. The grid shows 'Free' for all time slots across all days.

Time	Monday	Tuesday	Wednesday	Thursday
08:00	Free	Free	Free	Free
09:00	Free	Free	Free	Free
10:00	Free	Free	Free	Free
11:00	Free	Free	Free	Free
12:00	Free	Free	Free	Free
13:00	Free	Free	Free	Free
14:00	Free	Free	Free	Free
15:00	Free	Free	Free	Free
16:00	Free	Free	Free	Free
17:00	Free	Free	Free	Free

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Day	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Event Type	Lesson	Lesson	Lesson	Lesson	Lesson	Lesson	Lesson
Status	Free	Free	Free	Free	Free	Free	Free
Icon	Green	Green	Green	Green	Green	Green	Green

Developed by Thomas Alderson

Courses (/courses)

The screenshot shows a web application interface for 'Hubbal'. The top navigation bar includes a logo, user profile for 'Tom Alderson' (Last Login: 23/04/2016 - 11:23), and a menu icon. The main header 'Courses' is displayed with a subtitle 'Subjects of teaching.' Below the header, a sidebar lists navigation options: Main Website, Dashboard, Messages, Users, Groups (1), Timetable, and Courses/Classes. The main content area shows a list item for 'Website Development' under 'Head of Course' (Tom Alderson). The background features a large dark blue rectangular area.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Developed by Thomas Alderson

Appendix S: Frontend Design User Persona Example



Daniel Goldman
Professional Journalist

Daniel Goldman is a professional journalist, working for a local newspaper he works from home whilst writing, and is given his stories to write about via e-mail, he has a family of five and loves spending weekends away with them.

Interests

- Rock Climbing
- Fitness
- Music/Festivals
- Social Drinking

Goals & Needs

Daniel currently enjoys his writing career but would like to branch out into writing novel's and creating his own fantasy book series aimed at children.

Daniel also plans on having more children over the next few years.

Scenario

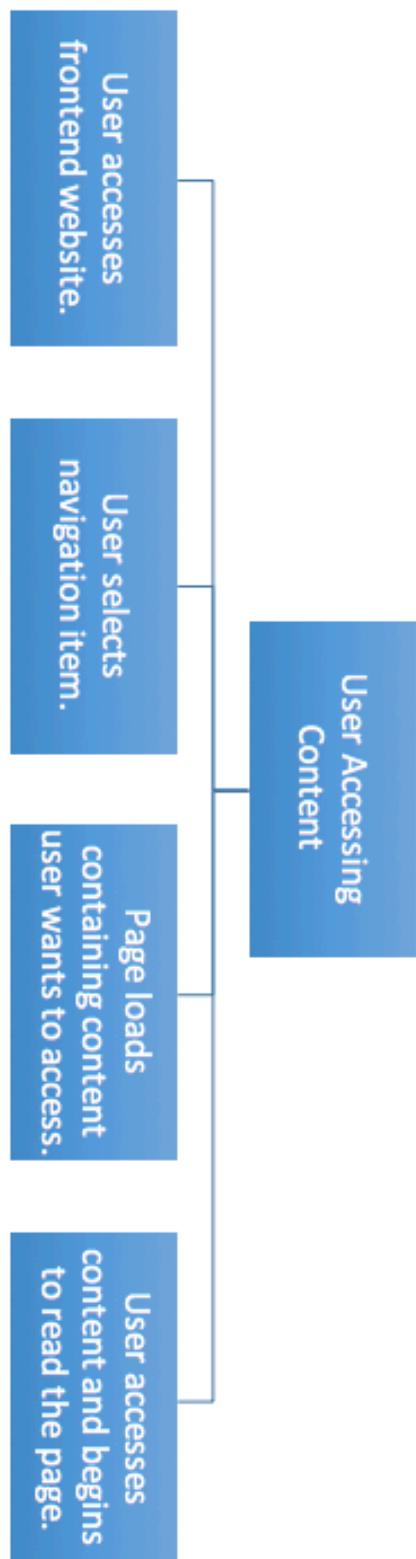
Daniel is interested in the local schools in the area and would like to be able to also view the available courses the school has to offer.

While he understands that school courses and classes can change at any time with the curriculum, he'd like to use a school website that is updated as soon as possible for convenience of getting the information he wants and being able to see it change the moment it is updated.

Appendix T: Frontend Design Hierarchical Task Analysis'

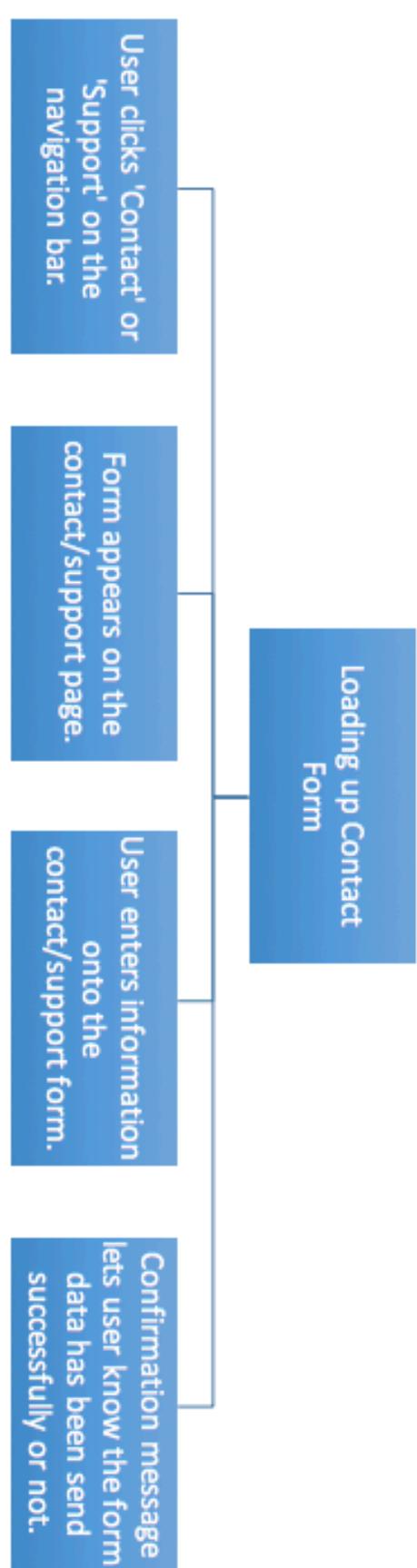
Hierarchical Task Analysis on User Content Access

This hierarchical task analysis will detail the user task map that will carry the user towards accessing specific content on the frontend website, this will not always be the same as other frontend designs because the frontend design is flexible and changes to the organisation the application is installed for.



Hierarchical Task Analysis on Accessing Contact Form

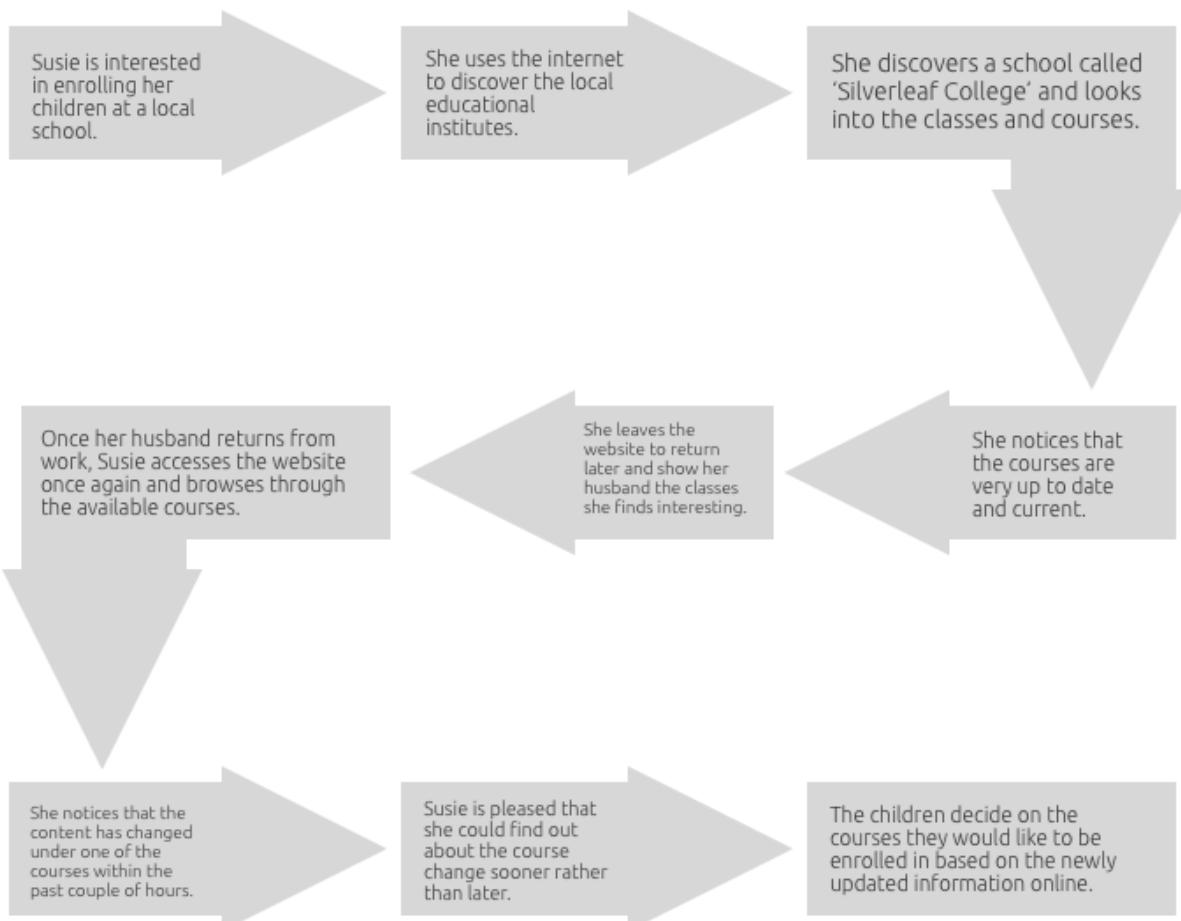
This hierarchical task analysis will demonstrate the task steps the user needs to take in order to successfully send a message to the administrators of the organisation that has the ~~Hubbal~~ application installed.



Appendix U: Frontend Design User Journey

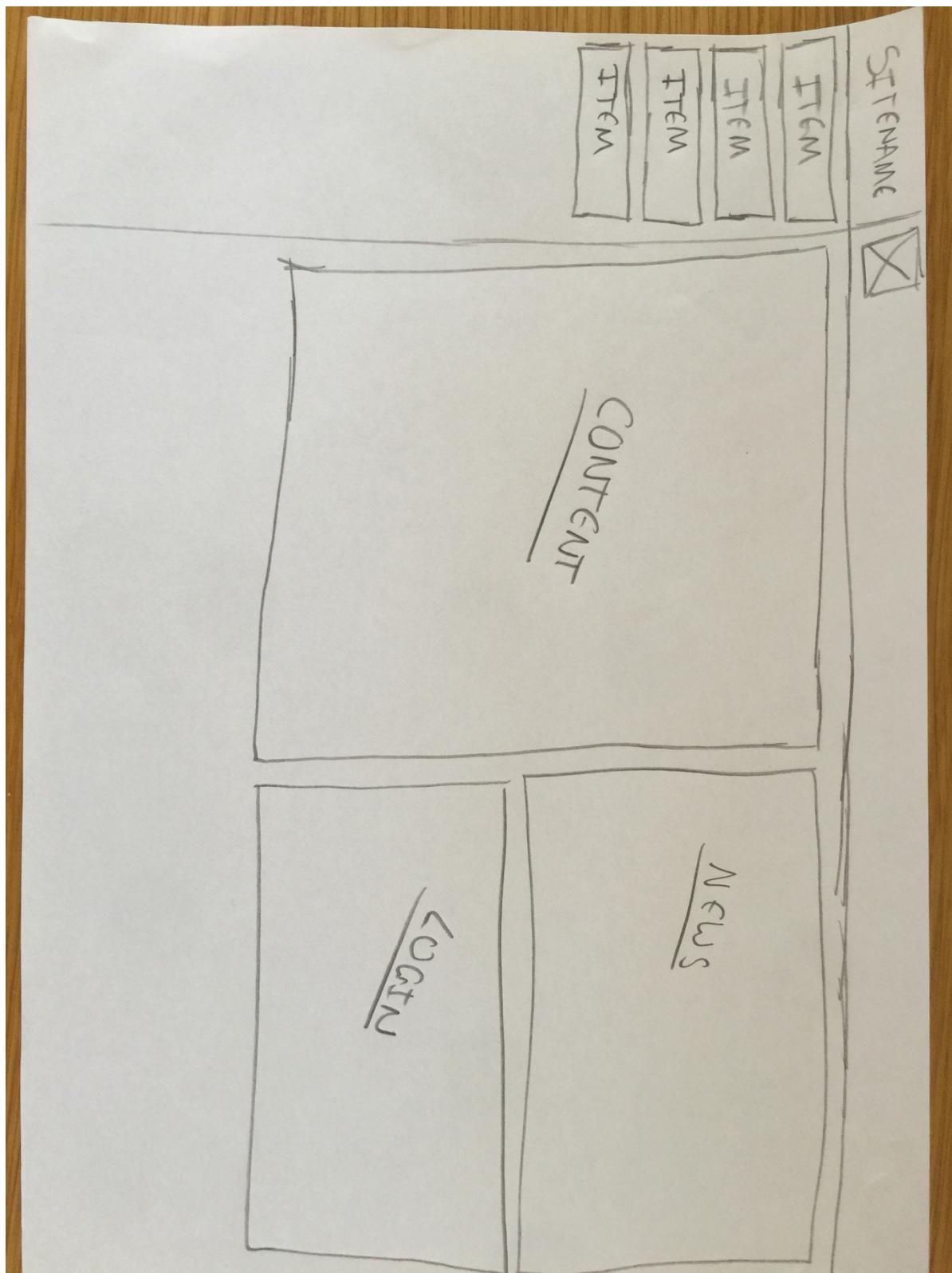
User Journey Frontend Design

The User Journey of the Frontend Design is flexible due to the organisation that the frontend design is being created for, this document is used to generate an idea of how to design a template based on what the users intended journey is from start to finish.

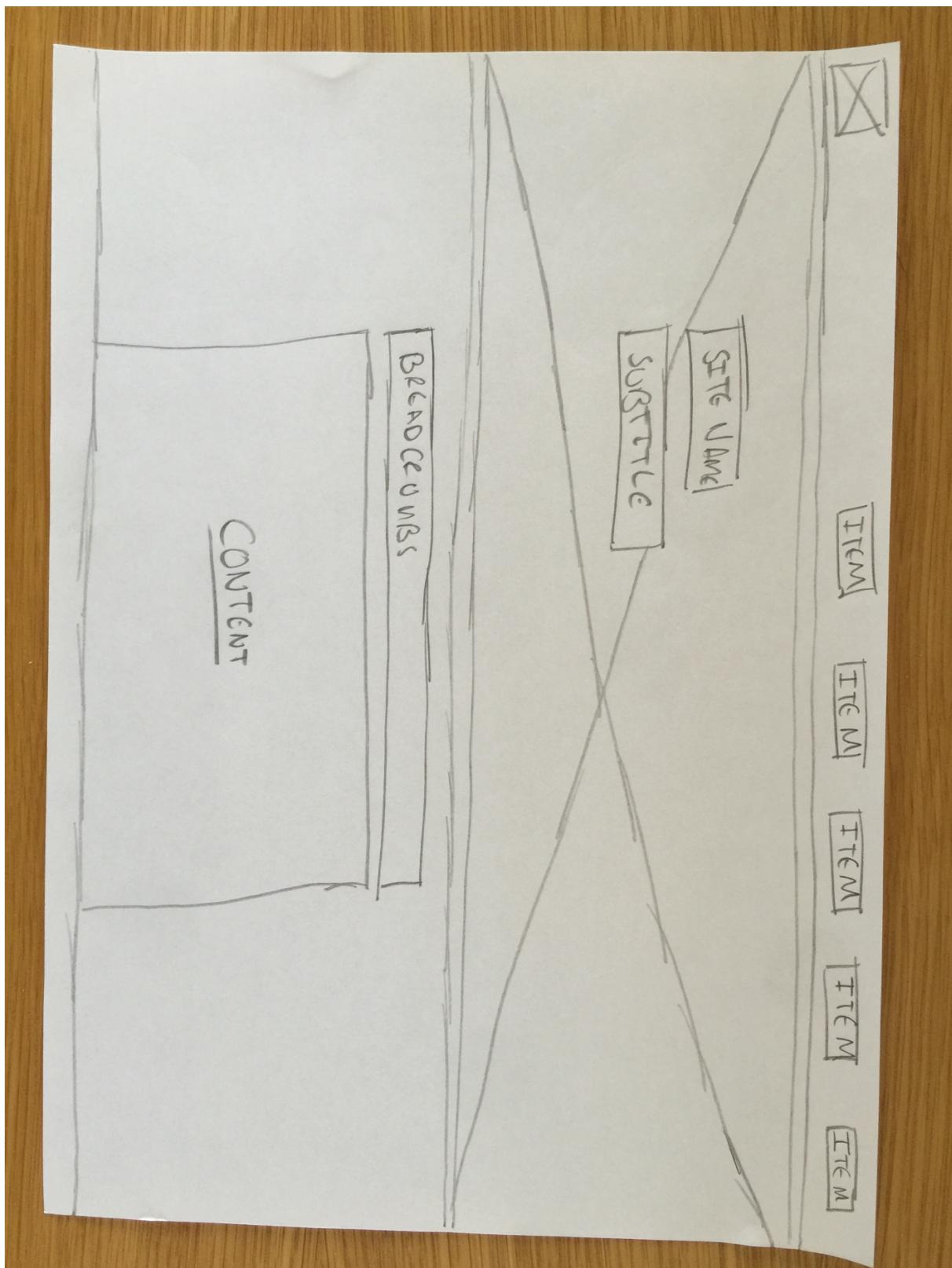


Appendix V: Low Fidelity Frontend Designs

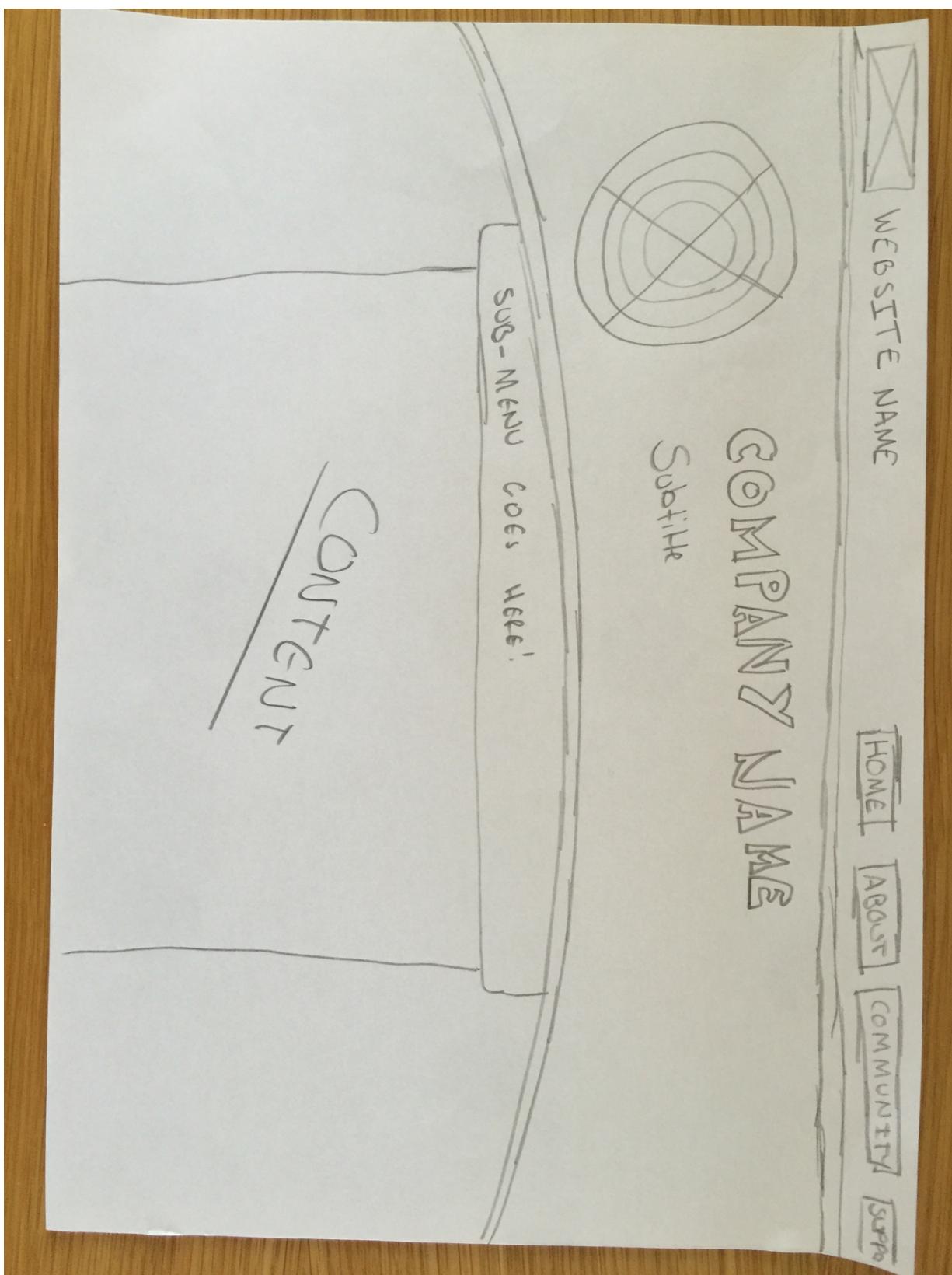
Design One



Design Two

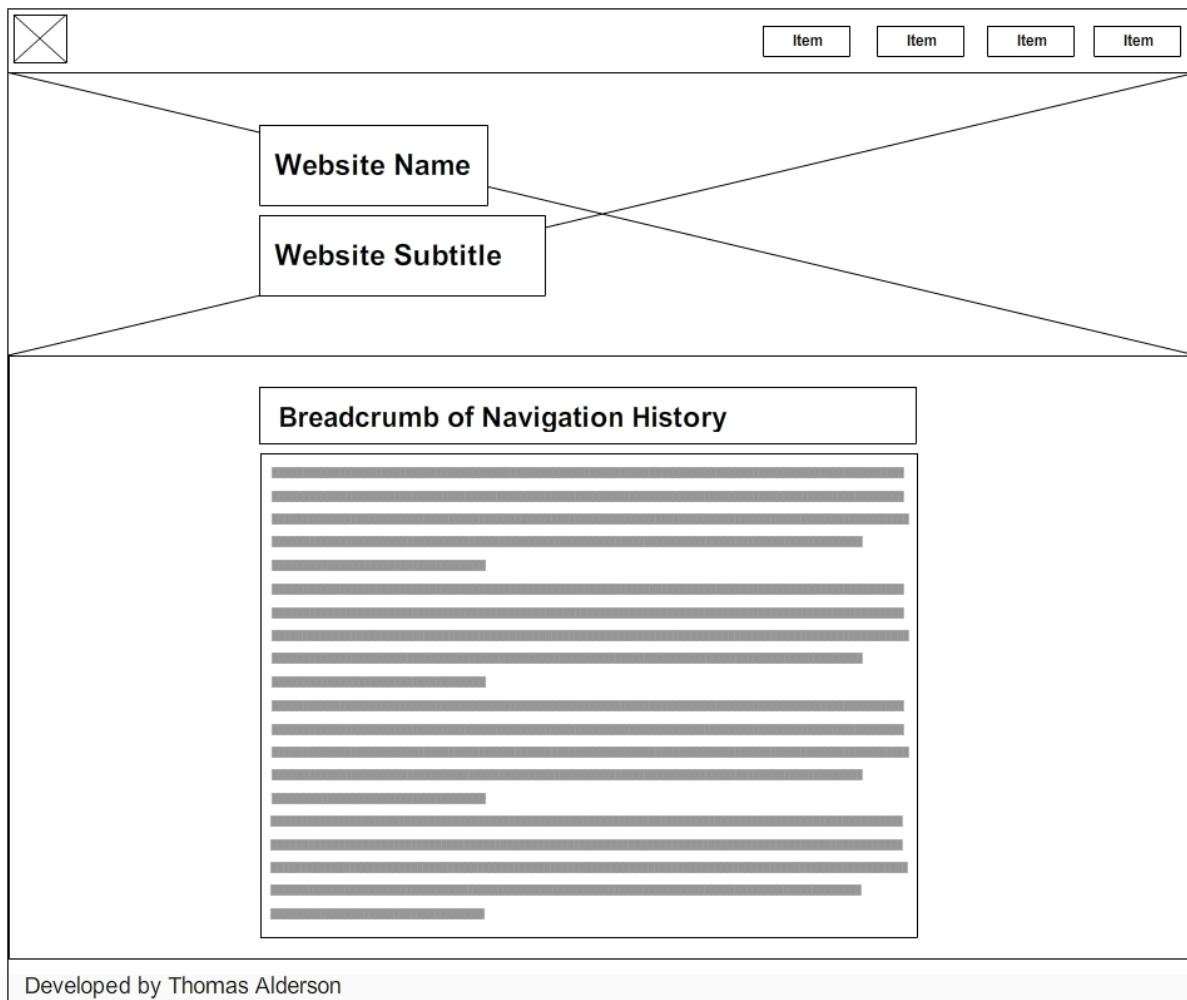


Design Three

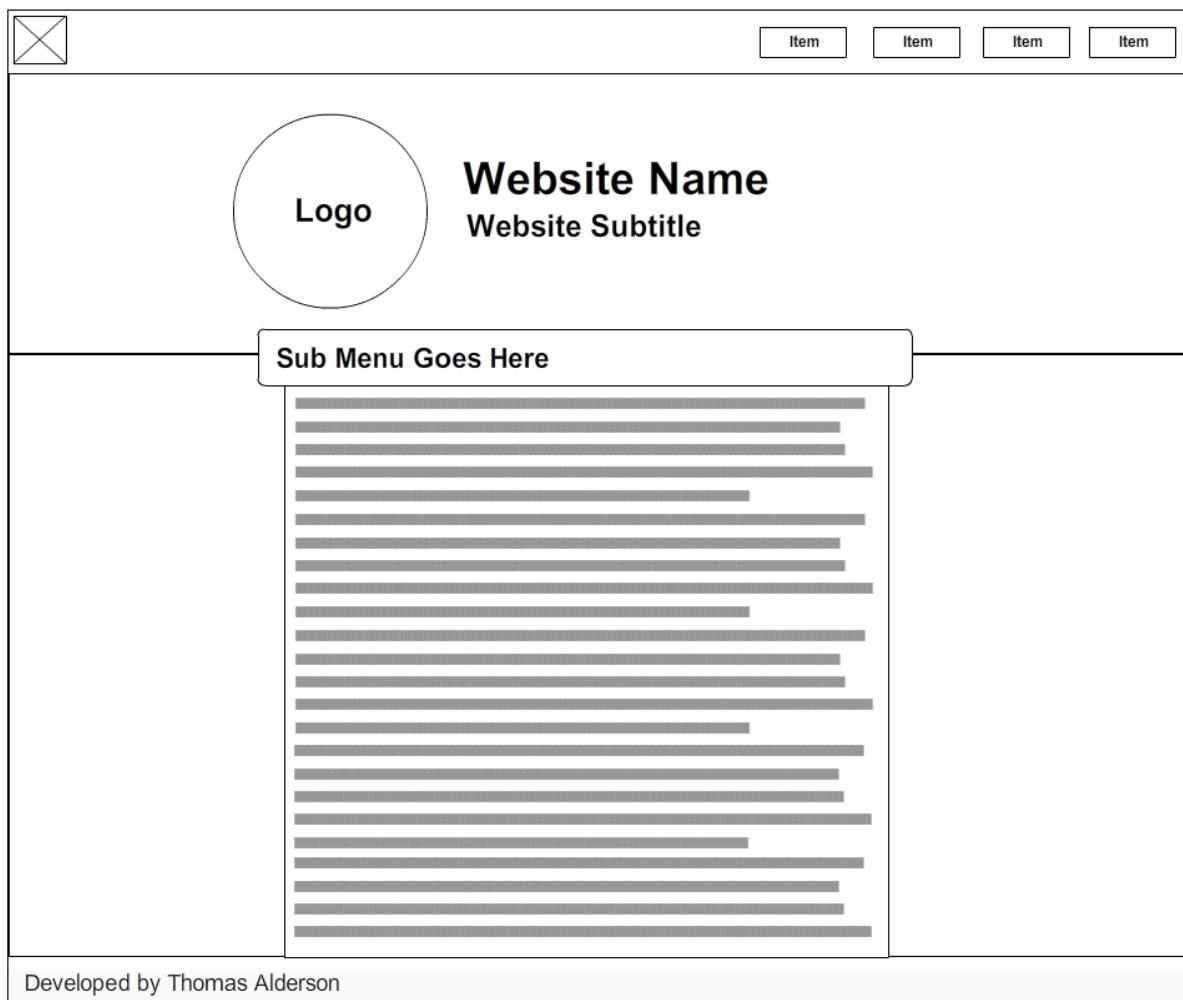


Appendix W: Mid Fidelity Frontend Designs

Design One



Design Two



Appendix X: High Fidelity Designs

Website developed for testing purposes of the Hubbal application. Silverleaf College is pure fictional and is in no way related to the Hubbal application besides testing.

Home

Why SilverLeaf College?
Silver Leaf College is known for its prestigious results from Ofsted and we pride ourselves on our reputation, in the past five years we have continued to deliver forever-improving results in GCSE exams.

How do I enrol my child?
Enrolment here at Silverleaf College is a simple process, requiring a simple phone call to request a prospectus and application form, from there all you need to do is fill in and return the completed forms.

What courses are available?
You will be able to find all of our available courses to enrol on in the 'Courses' section located in the navigation, alternatively, you can click [here](#).



Home About History Courses Location Support

Appendix Y: w3 Website Implementation Plan



The W3C Web Accessibility Initiative (WAI) Home page. The header features the W3C logo and the text "Web Accessibility Initiative". Below the header is a navigation menu with links to "Web Accessibility Initiative (WAI) Home", "Getting Started", "Designing for Inclusion", "Guidelines & Techniques", "Planning & Implementing", and "Policy Resources". A yellow callout box contains the text: "Discover new resources for people with disabilities, policy makers, managers, and you!".

This guide outlines activities to help you integrate accessibility throughout the web production process. This applies to individual projects carried out in sequence, and are ideally repeated over time to continually raise the level of accessibility capability.

A companion document, *Improving the Accessibility of Your Website*, provides guidance on more immediate remediation of accessibility issues.

Initiate

- Develop understanding of accessibility and build organizational enthusiasm.
 - Learn the basics
 - Explore the current environment
 - Set objectives
 - Develop business case
 - Raise awareness
 - Gather support
 - Engage with stakeholders

Plan

- Develop clear goals and an environment that supports accessibility.
 - Create accessibility policy
 - Assign responsibilities
 - Determine budget and resources
 - Review environment
 - Review websites
 - Establish monitoring framework
 - Track and communicate progress

Implement

- Ensure personnel are trained, and accessibility is included in their roles.
 - Build skills and expertise
 - Integrate goals into policies
 - Assign tasks and support development
 - Evaluate early and regularly
 - Prioritize issues
 - Track and communicate progress

Help improve this page

Please share your ideas, suggestions, or comments via email to the publicly-archived list wai-editioners@w3.org.

E-mail Fork & edit on GitHub New GitHub Issue

Status: Updated 31 March 2016 (first published October 2002)
Editors: Kevin White, Shadi Abou-Zahra, and Shawn Lawton Henry. Acknowledgements includes notes on past iterations. Developed by the Education and Outreach Working Group (EOWG). Developed with support from the WAI-DEV project.
[WAI Site Map] [Help with WAI Website] [Search] [Contacting WAI]
Feedback welcome to wai-editioners@w3.org (a publicly archived list) or wai@w3.org (a WAI staff-only list).

Copyright © 2016 W3C. (MIT, ERCIM, Keio, Beihang) Usage policies apply.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Skip to Content | Change text size or colors

ideelines, resources to make the Web accessible to people with disabilities

ts and on an organizational level. These activities are not necessarily barriers in existing websites.



tools are available, throughout, processes, and resources.

- Continue to review and report on content, delivery
- Monitor websites
- Engage with stakeholders
- Track standards and legislation
- Adapt to new technologies
- Incorporate user feedback

editors@w3.org or via GitHub.

, co-funded by the European Commission IST Programme.

Appendix Z: Message Board Discussion of Using iMac as a Web Development Focused Computer

The screenshot shows a forum thread on the MacRumors website. The thread is titled "iMac or Mac Mini for Web Development?" and was started by "Mooey" on Oct 21, 2012. The thread has five posts:

- Mooey (#1)** posted on Oct 21, 2012, asking if it would be better to get an iMac or a Mac Mini for web development. They mention needing Adobe Photoshop/Fireworks and some programming. Response: "with the upcoming release, I was wondering if it would be better to get an iMac or a Mac Mini for just mainly web development? I will probably have to have Adobe Photoshop / Fireworks installed, and I may do a bit of programming outside of the web development workflow. I was wondering if the Mac Mini would suffice, or if I would just need to bump up to the iMac."
Thanks.
- 12dylan34 (#2)** posted on Oct 21, 2012, responding that the mini with an appropriately large screen would suffice, unless you intend to game as well, get the iMac.
The mini with an appropriately large screen would suffice, I think. It comes down to what else you want to do with the machine. If you intend to game as well, get the iMac.
- Mooey (#3)** posted on Oct 21, 2012, responding that it's for work primarily so they won't be doing any gaming or anything, just purely web development and some front-end design with Photoshop and Fireworks.
12dylan34 said: ↑
The mini with an appropriately large screen would suffice, I think. It comes down to what else you want to do with the machine. If you intend to game as well, get the iMac.
It's for work primarily so I won't be doing any gaming or anything, just purely web development and some front-end design with Photoshop and Fireworks.
- fig (#4)** posted on Oct 22, 2012, responding that you should be fine with the Mini if that's the route you choose to go. They use a much slower machine now for web dev, design, and photoshop work and don't have any issues unless they get into high res print work.
Yeah, you should be fine with the Mini if that's the route you choose to go. I use a much slower machine now for web dev, design, and photoshop work and don't have any issues unless I get into high res print work.
Just load it up with RAM and go with the SSD (or add one) if possible and you'll have a snappy little machine.
- driftless (#5)** posted on Oct 22, 2012, responding that the iMac is a better machine. Price them both out after the updates are announced tomorrow.
The iMac is a better machine. Price them both out after the updates are announced tomorrow.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Oct 22, 2012

segovius
macrumors regular



I am in the same boat. My plan is simple yet cunning:
IF iMac = matte option or less glare revised screen
THEN iMac
IF ELSE
Mac Mini with 3rd party matte screen

SHARE | LIKE | REPLY

Oct 22, 2012

Geolink
macrumors member



Lol!

SHARE | LIKE | REPLY

Oct 22, 2012

phoenixsan
macrumors 65816



I will go with....
The iMac, because:
1- In the 27 inch version you have plenty of screen to work
2-Better graphics
3-Better storage options
4-Slightly better processor
If so, I will modify the iMac with more RAM (max out) or maybe a SSD
Having say the above, if you have a nice screen, good keyboard and mouse, maybe the mini is the way to go, with modifications:
-SSD
-Max out the RAM
-External Firewire storage
😊

SHARE | LIKE | REPLY

Oct 23, 2012

Mooey
thread starter
macrumors regular

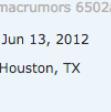


Now that both the Mac Mini and iMac have been announced, can anyone give me an opinion on what they would do if they were in my position?

SHARE | LIKE | REPLY

Oct 23, 2012

fig
macrumors 6502a



Mooey said: ↑
Now that both the Mac Mini and iMac have been announced, can anyone give me an opinion on what they would do if they were in my position?
I'm in a similar boat and it's honestly a coin flip, probably less so if you already have a really nice display. Both will be more than adequate, I'm going to check out the benchmarks when everything is released and then decide.

SHARE | LIKE | REPLY

Oct 23, 2012

driftless
macrumors 65816



iMac. No question.

SHARE | LIKE | REPLY

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Chicago-area

Oct 23, 2012

BillyBobBongo
macrumors 6800

Jun 21, 2007
On The Interweb Thingy!

Mooey said: ↑
Now that both the Mac Mini and iMac have been announced, can anyone give me an opinion on what they would do if they were in my position?

I'd go with the iMac.
Keep in mind that the only program that is really going to tax your system, that you mentioned, is Photoshop. The base model with 8GB of RAM will easily suit your needs, and no doubt serve you well for the next few years without ever requiring an upgrade.

#12

Oct 23, 2012

SrWebDeveloper
macrumors 68000

Dec 7, 2007
Alexandria, VA, USA

Both have sufficient resources (memory, wifi, ethernet, disk storage) to do what you do since you noted you have the peripherals and monitors. One is obviously much more portable if that matters to you, so maybe you need to figure out of being able to toss it in your soft case and connect remotely from home (if you have monitor, peripherals there as well) or you don't care and always work in your office.
I personally might consider the retina capable MacBook Pro 13" or 15" as I get complete portability and remote connectivity along with ability to dual monitor at work or home. Especially if you travel or do presentations.

#13

Oct 23, 2012

blanka
macrumors 6800
Jul 30, 2012

Go with the Mini. The iMac is a total new design that is going to have a false start with production problems. With the mini, you're set. Attach it to a Dell 2711 or the new 27 inch samsung PLS displays, and you have a great matte 2560x1440 display that in the Dell case also allows AdobeRGB/sRGB switching (Macs are limited to sRGB only with white-LED backlights).

#14

Oct 23, 2012

NVWebsites
macrumors newbie
Oct 20, 2012
Canada

I think you will be covered with either. I have a Mac Pro and looked at the specs today of the iMac; it looks relatively comparable except I opted for the 8 core- 2 X 2.4 GHz Quad.
I run Adobe Design Premium products like Dreamweaver and Illustrator. I use Eclipse to write PHP on my localhost.
I'm very pleased with how it runs. I use Mackeeper periodically to clear all the cache that is generated when I feel like it needs a bit of cleanup.

#15

Oct 24, 2012

LaunchpadBS
macrumors 6502a
Nov 11, 2008
iLondon/iDurban

The new Mac Mini base model even should more than cover you, I'm doing a fair amount of dev work on my 13" MBP base model with 16GB ram and an SSD and even with XCode, MSVS2008 and GIMP I have no hassles running iTunes and 3 browsers for testing.
People seem to be massively underestimating these new gen intel processors.
At the office they have us working on 4 year old core 2 duo machines and I struggle to even slow those down with all sorts of dev tools and DB's running.

#16

Oct 24, 2012

GGJstudios
macrumors Westmere

May 16, 2008

Mooey said: ↑
My question is which would be better for what I do at work?
Any Mac model can easily handle web development tasks.

NVWebsites said: ↑
I use Mackeeper periodically to clear all the cache that is generated when I feel like it needs a bit of cleanup.

I highly recommend you uninstall MacKeeper. You don't need "cleaner" or "maintenance" apps to keep your Mac running well, and some of these apps can do more harm than good. Some can even degrade, rather than improve

#17

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

running well, and some of these apps can do more harm than good. Some can even degrade, rather than improve system performance.

Some remove files/folders or unused languages or architectures, which does nothing more than free up some drive space, with the risk of deleting something important in the process. These apps will not make your Mac run faster or more efficiently, since having stuff stored on a drive does not impact performance, unless you're running out of drive space.

Some of these apps delete caches, which can hurt performance, rather than help it, since more system resources are used and performance suffers while each cache is being rebuilt. **Caches exist to improve performance, so deleting them isn't advisable in most cases.**

Many of the tasks performed by these apps should only be done selectively to troubleshoot specific problems, not en masse as routine maintenance.

Mac OS X does a good job of taking care of itself, without the need for 3rd party software. Among other things, it has its own maintenance scripts that run silently in the background on a daily, weekly and monthly basis, without user intervention.

Five Mac maintenance myths

Oct 24, 2012

fig macrumors 6502a Jun 13, 2012 Houston, TX

I think I commented on this in the other thread, but I now do more than what you're looking to do on a 2007 iMac. I think you're good either way 😊 #18

Oct 25, 2012

definitive macrumors 68000 Aug 4, 2008

13" macbook pro or air, and some ~\$200 24" or 27" screen with some converter plug should probably be enough for you. #19

Mar 10, 2013

Mooey thread starter macrumors regular Feb 8, 2007

Update:
Hey all,
I just wanted to give you all an update in case anyone was in the same boat. I went with the Mac Mini. It's perfect for everything I do. I bumped the RAM up to 16GB and threw in a SSD. It loads 50MB photoshop files flawlessly, compiles my code without a hitch, and uploads/downloads 1GB+ folders / files with no hiccups.
Perfect little machine for what I needed. I really liked the iMac's design, but I already had all of the equipment to make a computer. I would say if you don't have any monitors / keyboards / mice to go with the iMac, but if you have the other stuff, go with the Mac Mini. #20

Mar 13, 2013

tekboi macrumors 6502a Aug 9, 2006 New Orleans → Westward

I would have recommend the 27" iMAC for web design. This thing is an absolute beauty to work with. And the resolution is twice of most monitors of the same size. But if you already have monitors... I guess the mac mini would suffice. #21

(You must log in or sign up to post here.)

Share This Page

Twitter Share Google+

Forums > Special Interests > Web Design and Development

Appendix AA: Ruby & Rails Command Line Cheatsheet

Console ▾ Ruby ▾ Rails Editor Tips Help

/ Cheat Sheet Conventions

Bold words are what is really important e.g. the command and concept shown in the usage category. In the code usage and example columns these highlight the main part of the concept, like this: `general_stuff.concept`. In the same columns *italic_words* mark the arguments/parameters of a command/method.

However *italic words* in the descriptions or general text denote more general concepts or concepts explained elsewhere in this cheat sheet or in general.

/ Console Basics

The console (also called *command line*, *command prompt* or *terminal*) is just another way of interacting with your computer. So you can basically do anything with it that you could also do with your graphical desktop user interface. This section contains a couple of examples.

For the different operating systems **starting the console** differs.

- **Windows:** Open the start menu and search for command prompt. Alternatively choose execute and enter cmd.
- **Mac:** Open Spotlight, type terminal, and start that program.
- **Linux:** The terminal should be one of the main options once you open the main menu of your distribution. Otherwise search for terminal if your distribution has such an option or look under Accessories.

Concept	Usage	Examples	Description
Change directory	<code>cd directory</code>	<code>cd my_app</code> <code>cd my_app/app/controllers</code>	Changes the directory to the specified directory on the console.
List contents directory	<code>ls directory</code> Windows: <code>dir directory</code>	<code>ls</code> <code>ls my_app</code>	Shows all contents (files and folders) of the directory. If no directory is specified shows the contents of the current directory.
Directory you are currently in	<code>pwd</code>	<code>pwd</code>	Shows the full path of the directory you are currently in. E.g. /home/tobi/railsgirls A note on filenames: if a file or directory name starts with a slash / as in the output of pwd above, it is an absolute filename specifying the complete filename starting at the root of the current file system (e.g. hard disk). If the slash (/) is omitted, the file name is relative to the current working directory.
Create a new directory	<code>mkdir name</code>	<code>mkdir rails</code> <code>mkdir fun</code>	Creates a directory with the given name in the folder you are currently in.
Delete a file	<code>rm file</code> Windows: <code>del file</code>	<code>rm foo</code> <code>rm index.html</code> <code>rm pictures/old_picture.jpg</code>	Deletes the specified file. Be extra cautious with this as it would be too bad to delete something you still need :-(You can simply specify the name of a file or the directory you are currently in. However you can also specify a path, this is shown in the third example. There we delete the old_picture.jpg file from the pictures folder.
Delete a directory	<code>rm -r folder</code> Windows: <code>rd folder</code>	<code>rm -r stuff_i_dont_need</code> <code>rm -r stuff_i_dont_need/</code> <code>rm -r old_application</code>	Deletes the specified folder and all of its contents . So please be super cautious with this! Make sure that you do not need any of the contents of this folder any more. So why would you want to delete a whole folder? Well maybe it was an old application that you do not need anymore :-)
Starting a program	<code>program arguments</code>	<code>firefox</code> <code>firefox railsgirlsberlin.de</code> <code>irb</code>	Starts the program with the given name and arbitrary arguments if the program takes arguments. Firefox is just one example. Starting Firefox without arguments just opens up Firefox. If you give it an argument it opens the specified URL. When you type in <code>irb</code> this starts <i>interactive ruby</i> .
Abort the program	Press <code>Ctrl + C</code>	-	This will abort the program currently running in the terminal. For instance this is used to shut down the Rails server. You can also abort many other related tasks with it, including: bundle install, rake db:migrate, git pull and many more!

/ Ruby Basics

Ruby is the programming language Ruby on Rails is written in. So most of the time you will be writing Ruby code. Therefore it is good to grasp the basics of Ruby. If you just want to play with Ruby, type `irb` into your console to start interactive ruby. There you can easily experiment with Ruby. To leave irb, type `exit`.

This is just a very small selection of concepts. This is especially true later on when we talk about what Arrays, Strings etc. can do. For more complete information have a look at [ruby-doc](#) or search with your favorite search engine!

// General concepts

Concept	Usage	Examples	Description
Comment	<code># Comment text</code>	<code># This text is a comment</code> <code>some.ruby_code # A comment</code>	Ruby ignores everything that is marked as a comment. It does not try to execute it. Comments are just there for you as information. Comments are also commonly used to <i>comment out code</i> . That is when you don't want some part of your code to execute

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Comment	<code># Comment text</code>	<code># This text is a comment</code> <code>some.ruby_code # A comment</code> <code># some.ignored_ruby_code</code>	Ruby ignores everything that is marked as a comment. It does not try to execute it. Comments are just there for you as information. Comments are also commonly used to <i>comment out code</i> . That is when you don't want some part of your code to execute but you don't want to delete it just yet, because you are trying different things out.
Variables	<code>variable = some_value</code>	<code>name = "Tobi"</code> <code>name # => "Tobi"</code>	With a variable you tell Ruby that from now on you want to refer to that value by the name you gave it. So for the first example, from now on when you use <code>name</code> Ruby will know that you meant <code>"Tobi"</code> .
Console output	<code>puts something</code>	<code>puts "Hello World"</code> <code>puts [1, 5, "mooo"]</code>	Prints its argument to the console. Can be used in Rails apps to print something in the console where the server is running.
Call a method	<code>object.method(arguments)</code>	<code>string.length</code> <code>array.delete_at(2)</code> <code>string.gsub("ae", "ä")</code>	Calling a method is also often referred to as <i>sending a message</i> in Ruby. Basically we are sending an object some kind of message and are waiting for its response. This message may have no arguments or multiple arguments, depending on the message. So we kindly ask the object to do something or give us some information. When you "call a method" or "send a message" something happens. In the first example we ask a String how long it is (how many characters it consists of). In the last example we substitute all occurrences of "ae" in the string with the German "ä".
Define a method	<code>def name(parameter)</code> <code># method body</code> <code>end</code>	<code>def greet(name)</code> <code>puts "Hi there " + name</code> <code>end</code>	Different kinds of objects (Strings, Numbers, Arrays...) understand different messages. Methods are basically reusable units of behaviour. And you can define them yourself just like this. Methods are small and focused on implementing a specific behaviour. Our example method is focused on greeting people. You could call it like this: <code>greet("Tobi")</code>
Equality	<code>object == other</code>	<code>true == true # => true</code> <code>3 == 4 # => false</code> <code>"Hello" == "Hello" # => true</code> <code>"Hello" == "Hello" # => false</code>	With two equal signs you can check if two things are the same. If so, <code>true</code> will be returned; otherwise, the result will be <code>false</code> .
Inequality	<code>object != other</code>	<code>true != true # => false</code> <code>3 != 4 # => true</code>	Inequality is the inverse to equality, e.g. it results in <code>true</code> when two values are not the same and it results in <code>false</code> when they are the same.
Decisions with if	<code>if condition</code> <code># happens when true</code> <code>else</code> <code># happens when false</code> <code>end</code>	<code>if input == password</code> <code>grant_access</code> <code>else</code> <code>deny_access</code> <code>end</code>	With if-clauses you can decide based upon a condition what to do. When the condition is considered true, then the code after it is executed. If it is considered false, the code after the "else" is executed. In the example, access is granted based upon the decision if a given input matches the password.
Constants	<code>CONSTANT = some_value</code>	<code>PI = 3.1415926535</code> <code>PI # => 3.1415926535</code>	Constants look like variables, just in UPCASE. Both hold values and give you a name to refer to those values. However while the value a variable holds may change or might be of an unknown value (if you save user input in a variable) constants are different. They have a known value that should never change. Think of it a bit like mathematical or physical constants. These don't change, they always refer to the same value.

// Numbers

Numbers are what you would expect them to be, normal numbers that you use to perform basic math operations.

More information about numbers can be found in the [ruby-doc of Numeric](#).

Concept	Usage	Examples	Description
normal Number	<code>number_of_your_choice</code>	<code>0</code> <code>-11</code> <code>42</code>	Numbers are natural for Ruby, you just have to enter them!
Decimals	<code>main.decimal</code>	<code>3.2</code> <code>-5.0</code>	You can achieve decimal numbers in Ruby simply by adding a point.
Basic Math	<code>n operator m</code>	<code>2 + 3 # => 5</code> <code>5 - 7 # => -2</code> <code>8 * 7 # => 56</code> <code>84 / 4 # => 21</code>	In Ruby you can easily use basic math operations. In that sense you may use Ruby as a super-powered calculator.
Comparison	<code>n operator m</code>	<code>12 > 3 # => true</code> <code>12 < 3 # => false</code> <code>7 >= 7 # => true</code> <code>age >= 18 # true or false</code>	Numbers may be compared to determine if a number is bigger or smaller than another number. When you have the age of a person saved in the <code>age</code> variable you can see if that person is considered an adult in Germany:

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

// Strings

Strings are used to hold textual information. They may contain single characters, words, sentences or a whole book. However you may just think of them as an ordered collection of characters.

You can find out more about Strings at the [ruby-doc page about Strings](#).

Concept	Usage	Examples	Description
Create	'A string'	'Hello World' 'a' 'Just characters 129 _!\$%^' ''	A string is created by putting quotation marks around a character sequence. A Ruby style guide recommends using single quotes for simple strings.
Interpolation	"A string and an #{expression}"	"Email: #{user.email}" "The total is #{Z + Z}" "A simple string"	You can combine a string with a variable or Ruby expression using double quotation marks. This is called "interpolation." It is okay to use double quotation marks around a simple string, too.
Length	string.length	"Hello".length # => 5 "".length # => 0	You can send a string a message, asking it how long it is and it will respond with the number of characters it consists of. You could use this to check if the desired password of a user exceeds the required minimum length. Notice how we add a comment to show the expected result.
Concatenate	string + string2	"Hello " + "reader" # => "Hello reader" "a" + "b" + "c" # => "abc"	Concatenates two or more strings together and returns the result.
Substitute	string.gsub(a_string, substitute)	"Hae".gsub("ae", "ä") # => "Hä" "Hae".gsub("b", "ä") # => "Hae" "Greenie".gsub("e", "u") # => "Gruuniu"	gsub stands for "globally substitute". It substitutes all occurrences of a_string within the string with substitute .
Access	string[position]	"Hello"[1] # => "e"	Access the character at the given position in the string. Be aware that the first position is actually position 0.

// Arrays

An array is an ordered collection of items which is indexed by numbers. So an array contains multiple objects that are mostly related to each other. So what could you do? You could store a collection of the names of your favorite fruits and name it *fruits*.

This is just a small selection of things an Array can do. For more information have a look at the [ruby-doc for Array](#).

Concept	Usage	Examples	Description
Create	[contents]	[] ["Rails", "fun", 5]	Creates an Array, empty or with the specified contents.
Number of elements	array.size	[] .size # => 0 [1, 2, 3].size # => 3 ["foo", "bar"].size # => 2	Returns the number of elements in an Array.
Access	array[position]	array = ["hi", "foo", "bar"] array[0] # => "hi" array[2] # => "bar"	As an Array is a collection of different elements, you often want to access a single element of the Array. Arrays are indexed by numbers so you can use a number to access an individual element. Be aware that the numbering actually starts with "0" so the first element actually is the 0th. And the last element of a three element array is element number 2.
Adding an element	array << element	array = [1, 2, 3] array << 4 array # => [1, 2, 3, 4]	Adds the element to the end of the array, increasing the size of the array by one.
Assigning	array[number] = value	array = ["hi", "foo", "bar"] array[2] = "new" array # => ["hi", "foo", "new"]	Assigning new Array Values works a lot like accessing them; use an equals sign to set a new value. Voila! You changed an element of the array! Weeeuuuu!
Delete at index	array.delete_at(i)	array = [0, 14, 55, 79] array.delete_at(2) array # => [0, 14, 79]	Deletes the element of the array at the specified index. Remember that indexing starts at 0. If you specify an index larger than the number of elements in the array, nothing will happen.
Iterating	array.each do el ... end	persons.each do p puts p.name end numbers.each do n n = n * 2 end	"Iterating" means doing something for <i>each</i> element of the array. Code placed between <i>do</i> and <i>end</i> determines what is done to each element in the array. The first example prints the name of every person in the array to the console

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

```
numbers.each do |n| n = n + <end>
```

The first example prints the name of every person in the array to the console. The second example simply doubles every number of a given array.

// Hashes

Hashes associate a *key* to some *value*. You may then retrieve the value based upon its key. This construct is called a *dictionary* in other languages, which is appropriate because you use the key to "look up" a value, as you would look up a definition for a word in a dictionary. Each key must be unique for a given hash but values can be repeated.

Hashes can map from anything to anything! You can map from Strings to Numbers, Strings to Strings, Numbers to Booleans... and you can mix all of those! Although it is common that at least all the keys are of the same class. *Symbols* are especially common as keys. Symbols look like this: `:symbol`. A symbol is a colon followed by some characters. You can think of them as special strings that stand for (symbolize) something! We often use symbols because Ruby runs faster when we use symbols instead of strings.

Learn more about hashes at [ruby-doc](#).

Concept	Usage	Examples	Description
Creating	<code>{key => value}</code>	<code>{:hobby => "programming"} {:42 => "answer", "score" => 100, :name => "Tobi"}</code>	You create a hash by surrounding the key-value pairs with curly braces. The arrow always goes from the <i>key</i> to the <i>value</i> depicting the meaning: <i>"This key points to this value."</i> . Key-value pairs are then separated by commas.
Accessing	<code>hash[key]</code>	<code>hash = {key => "value"} hash[:key] # => "value" hash[foo] # => nil</code>	Accessing an entry in a hash looks a lot like accessing it in an <i>array</i> . However with a hash the key can be anything, not just numbers. If you try to access a key that does not exist, the value <code>nil</code> is returned, which is Ruby's way of saying "nothing", because if it doesn't recognize the key it can't return a value for it.
Assigning	<code>hash[key] = value</code>	<code>hash = {:a => "b"} hash[:key] = "value" hash # => {:a=>"b", :key=>"value"}</code>	Assigning values to a hash is similar to assigning values to an array. With a hash, the key can be a number or it can be a symbol, string, number... or anything, really!
Deleting	<code>hash.delete(key)</code>	<code>hash = {:a => "b", :b => 10} hash.delete(:a) hash # => {:b=>10}</code>	You can delete a specified key from the hash, so that the key and its value can not be accessed.

/ Rails Basics

This is an introduction to the basics of Rails. We look at the general structure of a Rails application and the important commands used in the terminal.

If you do not have Rails installed yet, there is a [well maintained guide by Daniel Kehoe](#) on how to install Rails on different platforms.

// The Structure of a Rails app

Here is an overview of all the folders of a new Rails application, outlining the purpose of each folder, and describing the most important files.

Name	Description
app	This folder contains your application. Therefore it is the most important folder in Ruby on Rails and it is worth digging into its subfolders. See the following rows.
app/assets	Assets basically are your front-end stuff. This folder contains <i>images</i> you use on your website, <i>javascripts</i> for all your fancy front-end interaction and <i>stylesheets</i> for all your CSS making your website absolutely beautiful.
app/controllers	The controllers subdirectory contains the controllers, which handle the requests from the users. It is often responsible for a single resource type, such as places, users or attendees. Controllers also tie together the <i>models</i> and the <i>views</i> .
app/helpers	Helpers are used to take care of logic that is needed in the views in order to keep the views clean of logic and reuse that logic in multiple views.
app/mailers	Functionality to send emails goes here.
app/models	The models subdirectory holds the classes that model the business logic of our application. It is concerned with the things our application is about. Often this is data, that is also saved in the database. Examples here are a Person, or a Place class with all their typical behaviour.
app/views	The views subdirectory contains the display templates that will be displayed to the user after a successful request. By default they are written in HTML with embedded ruby (<code>.html.erb</code>). The embedded ruby is used to insert data from the application. It is then converted to HTML and sent to the browser of the user. It has subdirectories for every resource of our application, e.g. places, persons. These subdirectories contain the associated view files. Files starting with an underscore (_) are called <i>partials</i> . Those are parts of a view which are reused in other views. A common example is <code>_form.html.erb</code> which contains the basic form for a given resource. It is used in the <code>new</code> and in the <code>edit</code> view since creating something and editing something looks pretty similar.
config	This directory contains the configuration files that your application will need, including your database configuration (in <code>database.yml</code>) and the particularly important <code>routes.rb</code> which decides how web requests are handled. The <code>routes.rb</code> file matches a given URL with the <code>controller</code> which will handle the request.
db	Contains a lot of <i>database</i> related files. Most importantly the <i>migrations</i> subdirectory, containing all your database migration files. Migrations set up your database structure, including the attributes of your models. With migrations you can add new attributes to existing models or create new models. So you could add the <code>favorite_color</code> attribute to your Person model so everyone can specify their favorite color.
doc	Contains the documentation you create for your application. Not too important when starting out.
lib	Short for library. Contains code you've developed that is used in your application and may be used elsewhere. For example, this might be code used to get specific information from Facebook.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

to get specific information from Facebook.	
log	See all the funny stuff that is written in the console where you started the Rails server? It is written to your <code>development.log</code> . Logs contain runtime information of your application. If an error happens, it will be recorded here.
public	Contains static files that do not contain Ruby code, such as error pages.
script	By default contains what is executed when you type in the <code>rails</code> command. Seldom of importance to beginners.
test	Contains the tests for your application. With tests you make sure that your application actually does what you think it does. This directory might also be called <code>spec</code> , if you are using the RSpec gem (an alternative testing framework).
vendor	A folder for software code provided by others ("libraries"). Most often, libraries are provided as <code>ruby gems</code> and installed using the <code>Gemfile</code> . If code is not available as a ruby gem then you should put it here. This might be the case for jQuery plugins. Probably won't be used that often in the beginning.
Gemfile	A file that specifies a list of gems that are required to run your application. Rails itself is a gem you will find listed in the <code>Gemfile</code> . Ruby gems are self-contained packages of code, more generally called libraries, that add functionality or features to your application. If you want to add a new gem to your application, add " <code>*gem gem_name*</code> " to your <code>Gemfile</code> , optionally specifying a version number. Save the file and then run <code>bundle install</code> to install the gem.
Gemfile.lock	This file specifies the exact versions of all gems you use. Because some gems depend on other gems, Ruby will install all you need automatically. The file also contains specific version numbers. It can be used to make sure that everyone within a team is working with the same versions of gems. The file is auto-generated. <i>Do not edit this file.</i>

// Important Rails commands

Here is a summary of important commands that can be used as you develop your Ruby on Rails app. You must be in the root directory of your project to run any of these commands (with the exception of the `rails new` command). The project or application root directory is the folder containing all the subfolders described above (app, config, etc.).

Concept	Usage	Description
Create a new app	<code>rails new name</code>	Create a new Ruby on Rails application with the given name here. This will give you the basic structure to immediately get started. After this command has successfully run your application is in a folder with the same name you gave the application. You have to <code>cd</code> into that folder.
Start the server	<code>rails server</code>	You have to start the server in order for your application to respond to your requests. Starting the server might take some time. When it is done, you can access your application under <code>localhost:3000</code> in the browser of your choice. In order to stop the server, go to the console where it is running and press <code>Ctrl + C</code>
Scaffolding	<code>rails generate scaffold name attribute:type</code>	The scaffold command magically generates all the common things needed for a new resource for you! This includes <code>controllers</code> , <code>models</code> and <code>views</code> . It also creates the following basic actions: create a new resource, edit a resource, show a resource, and delete a resource. That's all the basics you need. Take this example: <code>rails generate scaffold product name:string price:integer</code>
		Now you can create new products, edit them, view them and delete them if you don't need them anymore. Nothing stops you from creating a full fledged web shop now ;-)
Run migrations	<code>rake db:migrate</code>	When you add a new migration, for example by creating a new <code>scaffold</code> , the migration has to be applied to your database. The command is used to update your database.
Install dependencies	<code>bundle install</code>	If you just added a new gem to your <code>Gemfile</code> you should run <code>bundle install</code> to install it. Don't forget to restart your server afterwards!
Check dependencies	<code>bundle check</code>	Checks if the dependencies listed in <code>Gemfile</code> are satisfied by currently installed gems
Show existing routes	<code>rake routes</code>	Shows complete list of available routes in your application.

// ERB: Embedded Ruby

In your views (that is, under `app/views` in your Rails app) you will find `.html.erb` files. **ERB** stands for **E**mbedded **R**uby. This just means that Rails processes some special tags in those files and produces HTML output to send back to the user.

There are two ERB tags that you need to remember: `<%= ruby_code %>` and `<% ruby_code %>`. Notice that the difference is the = in the first tag.

Tag	Examples	Description
<code><%= %></code>	<code><%= @product.price %></code>	It runs the Ruby code and inserts the result to the HTML at that position. You can put <i>any kind of Ruby code</i> between <code><%= %></code> and <code><%= %></code> , for instance, <code><%= 9 * 3 %></code> will translate to <code>27</code> in the page that the user is viewing. However, typically this tag is used to display some data from a model, such as the price of a product, as shown in the example here.
<code><% %></code>	<code><% if user.admin? %> <p>Hello Admin!</p> <% end %></code>	The Ruby code between the delimiters <code><%</code> and <code>%></code> is run but the result will not be inserted at this point in the HTML. Therefore these tags are most commonly used for control flow structures such as an if statement in the example, or loops.

/ Editor tips

When you write code you will be using a text editor. Of course each text editor is different and configurable. Here are just some functions and their most general short cuts. All of them work in `Sublime Text 2`. Your editor may differ!

The shortcuts listed here are for Linux/Windows. On a Mac you will have to replace `Ctrl` with `Cmd`.

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?



When you write code you will be using a text editor. Of course each text editor is different and configurable. Here are just some functions and their most general short cuts. All of them work in [Sublime Text 2](#). Your editor may differ!

The shortcuts listed here are for Linux/Windows. On a Mac you will have to replace *Ctrl* with *Cmd*.

Function	Shortcut	Description
Save file	Ctrl + S	Saves the currently open file. If it was a new file you may also be asked where to save it.
Undo	Ctrl + Z	Undo the last change you made to the current file. Can be applied multiple times in succession to undo multiple changes.
Redo	Ctrl + Y <i>or Ctrl + Shift + Z</i>	Redo what you just undid with <i>undo</i> , can also be done multiple times.
Find in File	Ctrl + F	Search for a character sequence within the currently open file. Hit <i>Enter</i> to progress to the next match.
Find in all Files	Ctrl + Shift + F	Search for a character sequence in all files of the project.
Replace	Ctrl + H <i>or Ctrl + R</i>	Replace occurrences of the supplied character sequence with the other supplied character sequence. Useful when renaming something.
Copy	Ctrl + C	Copy the currently highlighted text into the clipboard.
Cut	Ctrl + X	Copy the highlighted text into the clipboard but delete it.
Paste	Ctrl + V	Insert whatever currently is in the clipboard (through <i>Copy</i> or <i>Cut</i>) at the current caret position. Can insert multiple times.
New File	Ctrl + N	Create a new empty file.
Search and open file	Ctrl + P	Search for a file giving part of its name (<i>fuzzy search</i>). Pressing <i>enter</i> will open the selected file.
Comment	Ctrl + /	Marks the selected text as a comment, which means that it will be ignored. Useful when you want to see how something behaves or looks without a specific section of code being run.

/ Help: What to do when things go wrong?

Things go wrong all the time. Don't worry, this happens to everyone. So keep calm. When you encounter an error, just google the error message. For best results, add the keywords "rails" or "ruby". Results from [stackoverflow.com](#) are often really helpful. Look for those! The most experienced developers do this frequently ;-).

Here are common mistakes with a little checklist:

- Have you run `rake db:migrate` to apply the newest database migrations?
- Have you really saved the file you just changed? Unsaved files are often marked in the editor via an asterisk or a point next to their name.
- If you just added a gem to the Gemfile, have you run `bundle install` to install it?
- If you just installed a gem, have you restarted the server?

Do you need more beginner friendly in depth information about Ruby on Rails? We have started to gather free tutorials and learning material on a [resources](#) page! Please give feedback about your favorite tutorials and lessons!

created by [Tobias Pfeiffer](#)

[About](#) · [Resources](#) · [Blog](#)

Appendix BB: 'rails new' Command Output

```
iMac:Dropbox Tom$ rails new hubbal
[  create README.rdoc
[  create Rakefile
[  create config.ru
[  create .gitignore
[  create Gemfile
[  create app
[  create app/assets/javascripts/application.js
[  create app/assets/stylesheets/application.css
[  create app/controllers/application_controller.rb
[  create app/helpers/application_helper.rb
[  create app/views/layouts/application.html.erb
[  create app/assets/images/.keep
[  create app/mailers/.keep
[  create app/models/.keep
[  create app/controllers/concerns/.keep
[  create app/models/concerns/.keep
[  create bin
[  create bin/bundle
[  create bin/rails
[  create bin/rake
[  create bin/setup
[  create config
[  create config/routes.rb
[  create config/application.rb
[  create config/environment.rb
[  create config/secrets.yml
[  create config/environments
[  create config/environments/development.rb
[  create config/environments/production.rb
[  create config/environments/test.rb
[  create config/initializers
[  create config/initializers/assets.rb
[  create config/initializers/backtrace_silencers.rb
[  create config/initializers/cookies_serializer.rb
[  create config/initializers/filter_parameter_logging.rb
[  create config/initializers/inflections.rb
[  create config/initializers/mime_types.rb
[  create config/initializers/session_store.rb
[  create config/initializers/wrap_parameters.rb
[  create config/locales
[  create config/locales/en.yml
[  create config/boot.rb
[  create config/database.yml
[  create db
[  create db/seeds.rb
[  create lib
[  create lib/tasks
[  create lib/tasks/.keep
[  create lib/assets
[  create lib/assets/.keep
[  create log
[  create log/.keep
[  create public
[  create public/404.html
[  create public/422.html
[  create public/500.html
[  create public/favicon.ico
[  create public/robots.txt
[  create test/fixtures
[  create test/fixtures/.keep
```

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

```
create  test/controllers
create  test/controllers/.keep
create  test/mailers
create  test/mailers/.keep
create  test/models
create  test/models/.keep
create  test/helpers
create  test/helpers/.keep
create  test/integration
create  test/integration/.keep
create  test/test_helper.rb
create  tmp/cache
create  tmp/cache/assets
create  vendor/assets/javascripts
create  vendor/assets/javascripts/.keep
create  vendor/assets/stylesheets
create  vendor/assets/stylesheets/.keep
  run  bundle install
Fetching gem metadata from https://rubygems.org/.....
Fetching version metadata from https://rubygems.org/...
Fetching dependency metadata from https://rubygems.org/...
Resolving dependencies.....
Installing rake 11.1.2
Using i18n 0.7.0
Using json 1.8.3
Installing minitest 5.8.4
Using thread_safe 0.3.5
Using tzinfo 1.2.2
Using activesupport 4.2.4
Using builder 3.2.2
Using erubis 2.7.0
Installing mini_portile2 2.0.0
Installing nokogiri 1.6.7.2 with native extensions
Using rails-deprecated_sanitizer 1.0.3
Using rails-dom-testing 1.0.7
Using loofah 2.0.3
Installing rails-html-sanitizer 1.0.3
Using actionview 4.2.4
Using rack 1.6.4
Using rack-test 0.6.3
Using actionpack 4.2.4
Using globalid 0.3.6
Using activejob 4.2.4
Installing mime-types-data 3.2016.0221
Installing mime-types 3.0
Installing mail 2.6.4
Using actionmailer 4.2.4
Using activemodel 4.2.4
Using arel 6.0.3
Using activerecord 4.2.4
Using debug_inspector 0.0.2
Using binding_of_caller 0.7.2
Using bundler 1.10.6
Installing byebug 8.2.4 with native extensions
Installing coffee-script-source 1.10.0
Using execjs 2.6.0
Using coffee-script 2.4.1
Using thor 0.19.1
Using railties 4.2.4
Installing coffee-rails 4.1.1
Installing concurrent-ruby 1.0.1
Installing multi_json 1.11.3
Installing jbuilder 2.4.1
Installing jquery-rails 4.1.1
Installing sprockets 3.6.0
```

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

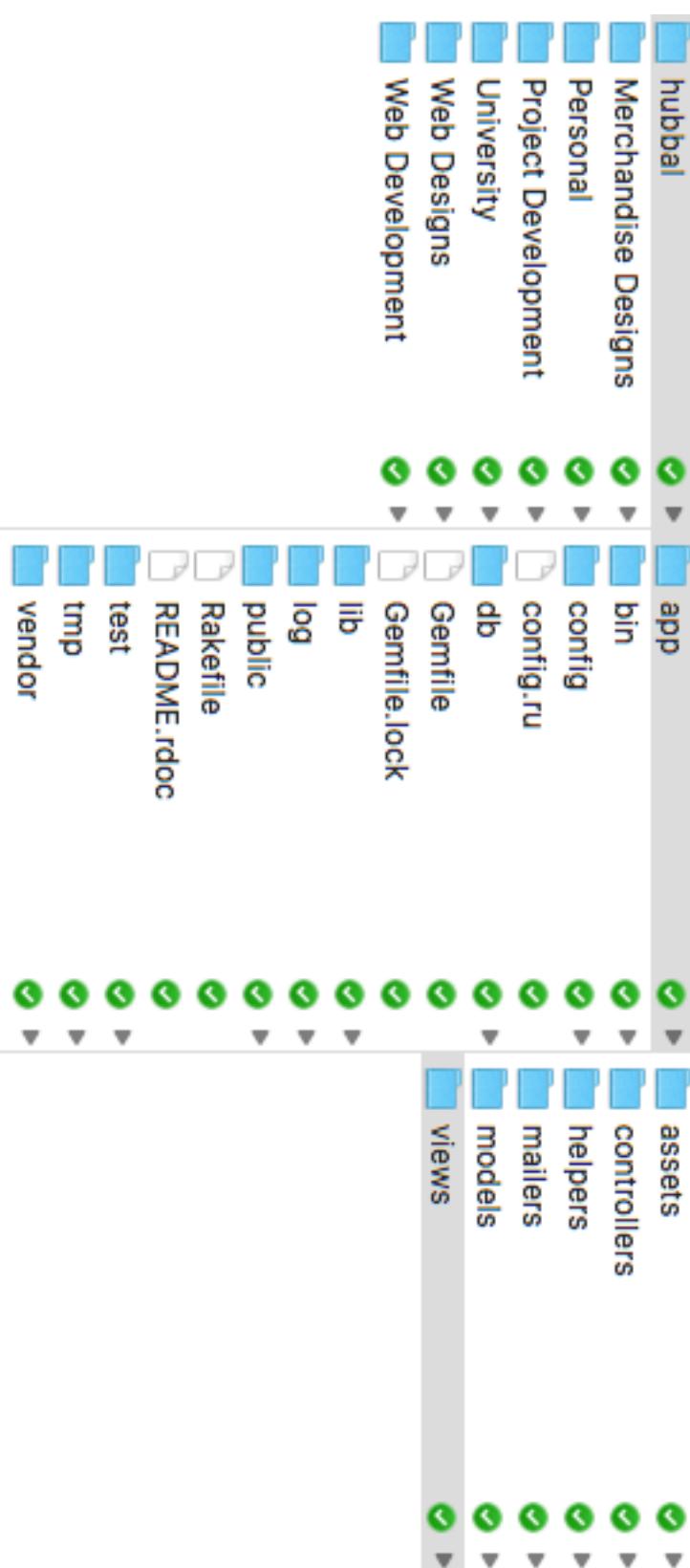
```
Installing sprockets-rails 3.0.4
Using rails 4.2.4
Installing rdoc 4.2.2
Installing sass 3.4.22
Installing tilt 2.0.2
Using sass-rails 5.0.4
Using sdoc 0.4.1
Installing spring 1.7.1
Installing sqlite3 1.3.11 with native extensions
Using turbolinks 2.5.3
Installing uglifier 3.0.0
Installing web-console 2.3.0
Bundle complete! 12 Gemfile dependencies, 55 gems now installed.
Use `bundle show [gemname]` to see where a bundled gem is installed.
Post-install message from rdoc:
Depending on your version of ruby, you may need to install ruby rdoc/ri data:

<= 1.8.6 : unsupported
= 1.8.7 : gem install rdoc-data; rdoc-data --install
= 1.9.1 : gem install rdoc-data; rdoc-data --install
>= 1.9.2 : nothing to do! Yay!
    run bundle exec spring binstub --all
* bin/rake: spring inserted
* bin/rails: spring inserted
```

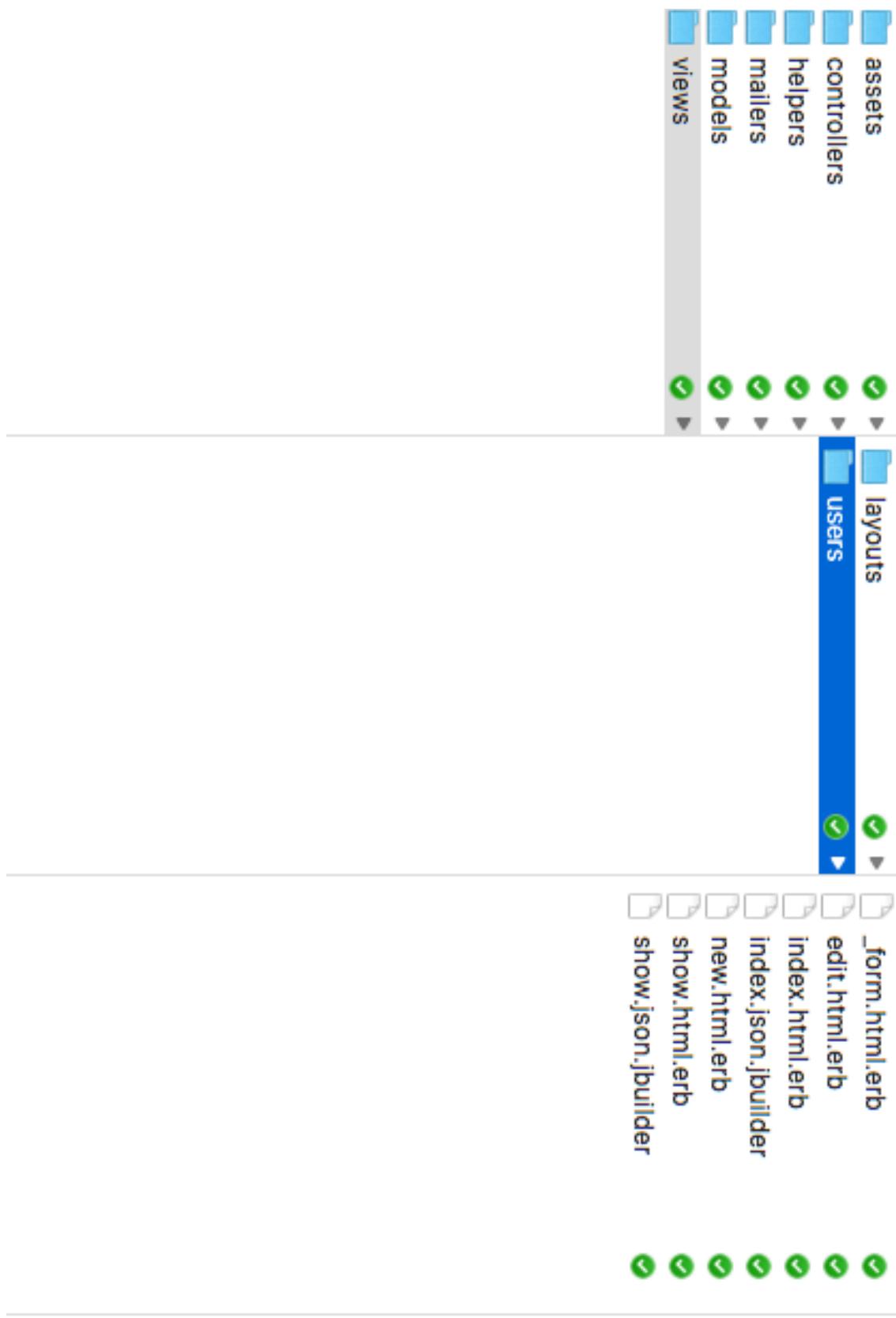
Appendix CC: 'rails generate scaffold User' Output

```
iMac:Dropbox Tom$ cd hubbal
iMac:hubbal Tom$ rails generate scaffold User firstname:string lastname:string username:string email:string password_digest:string phone:string role:integer status:integer room:string
Running via Spring preloader in process 563
  invoke  active_record
    create    db/migrate/20160426132516_create_users.rb
    create    app/models/user.rb
  invoke    test_unit
    create    test/models/user_test.rb
    create    test/fixtures/users.yml
  invoke    resource_route
    route      resources :users
  invoke scaffold_controller
    create    app/controllers/users_controller.rb
  invoke    erb
    create    app/views/users
    create    app/views/users/index.html.erb
    create    app/views/users/edit.html.erb
    create    app/views/users/show.html.erb
    create    app/views/users/new.html.erb
    create    app/views/users/_form.html.erb
  invoke    test_unit
    create    test/controllers/users_controller_test.rb
  invoke    helper
    create    app/helpers/users_helper.rb
  invoke    test_unit
  invoke    jbuilder
    create    app/views/users/index.json.jbuilder
    create    app/views/users/show.json.jbuilder
  invoke assets
  invoke    coffee
    create    app/assets/javascripts/users.coffee
  invoke scss
    create    app/assets/stylesheets/users.scss
  invoke scss
    create    app/assets/stylesheets/scaffolds.scss
```

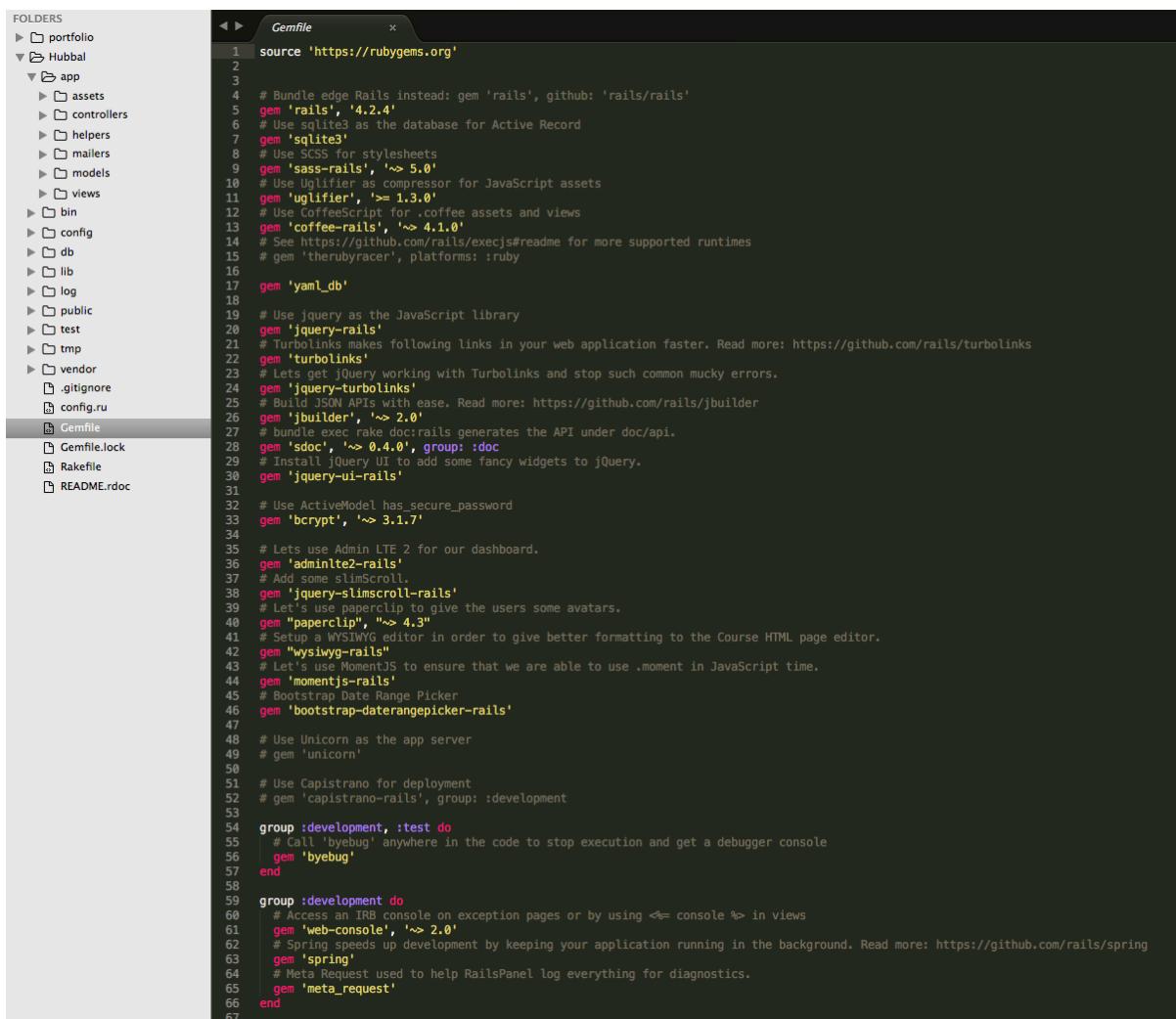
Appendix DD: Files Developed by Scaffold



CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?



Appendix EE: Application Gemfile



The screenshot shows a terminal window with a file tree on the left and the contents of the Gemfile on the right. The file tree includes folders like portfolio, Hubbal, app, config, db, lib, log, public, test, tmp, vendor, .gitignore, Rakefile, config.ru, and README.rdoc. The Gemfile itself contains numerous gem specifications and group definitions for development, test, and production environments.

```
source 'https://rubygems.org'

# Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
gem 'rails', '4.2.4'
# Use SQLite3 as the database for Active Record
gem 'sqlite3'
# Use SCSS for stylesheets
gem 'sass-rails', '>= 5.0'
# Use Uglifier as compressor for JavaScript assets
gem 'uglifier', '>= 1.3.0'
# Use CoffeeScript for .coffee assets and views
gem 'coffee-rails', '>= 4.1.0'
# See https://github.com/rails/execjs#readme for more supported runtimes
# gem 'therubyracer', platforms: :ruby

gem 'yaml_db'

# Use jquery as the JavaScript library
gem 'jquery-rails'
# Turbolinks makes following links in your web application faster. Read more: https://github.com/rails/turbolinks
gem 'turbolinks'
# Lets get jQuery working with Turbolinks and stop such common mucky errors.
gem 'jquery-turbolinks'
# Build JSON APIs with ease. Read more: https://github.com/rails/jbuilder
gem 'jbuilder', '>= 2.0'
# bundle exec rake doc:rails generates the API under doc/api.
gem 'sdoc', '>= 0.4.0', group: :doc
# Install jQuery UI to add some fancy widgets to jQuery.
gem 'jquery-ui-rails'

# Use ActiveModel has_secure_password
gem 'bcrypt', '>= 3.1.7'

# Lets use Admin LTE 2 for our dashboard.
gem 'adminlte2-rails'
# Add some slimScroll.
gem 'jquery-slimscroll-rails'
# Let's use paperclip to give the users some avatars.
gem 'paperclip', '>= 4.3'
# Setup a WYSIWYG editor in order to give better formatting to the Course HTML page editor.
gem "wysiwyg-rails"
# Let's use MomentJS to ensure that we are able to use .moment in JavaScript time.
gem 'momentjs-rails'
# Bootstrap Date Range Picker
gem 'bootstrap-daterangepicker-rails'

# Use Unicorn as the app server
# gem 'unicorn'

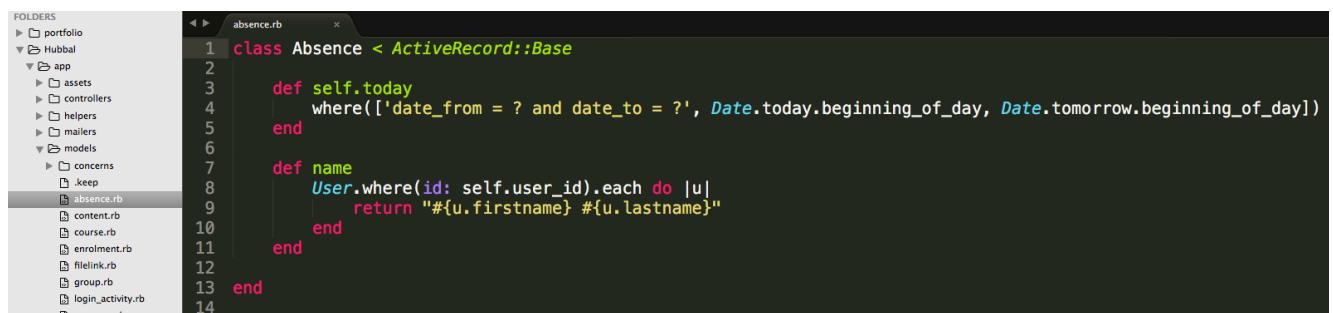
# Use Capistrano for deployment
# gem 'capistrano-rails', group: :development

group :development, :test do
  # Call 'byebug' anywhere in the code to stop execution and get a debugger console
  gem 'byebug'
end

group :development do
  # Access an IRB console on exception pages or by using <code>console</code> in views
  gem 'web-console', '>= 2.0'
  # Spring speeds up development by keeping your application running in the background. Read more: https://github.com/rails/spring
  gem 'spring'
  # Meta Request used to help RailsPanel log everything for diagnostics.
  gem 'meta_request'
end
```

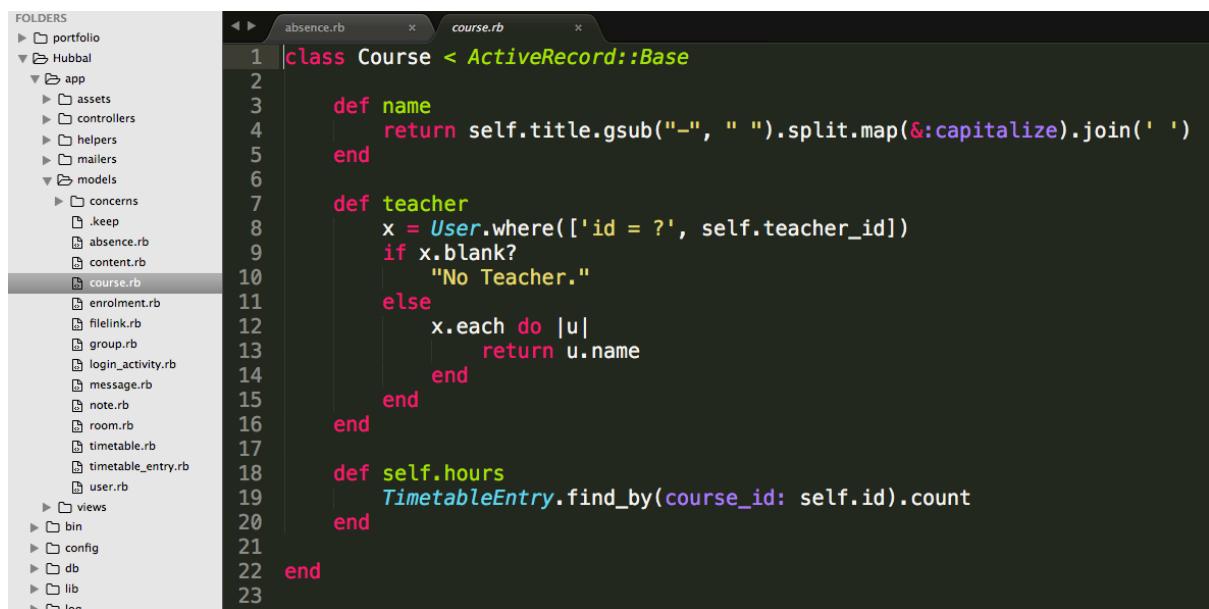
Appendix FF: Application Model Files

absence.rb



```
1 class Absence < ActiveRecord::Base
2
3   def self.today
4     where(['date_from = ? and date_to = ?', Date.today.beginning_of_day, Date.tomorrow.beginning_of_day])
5   end
6
7   def name
8     User.where(id: self.user_id).each do |u|
9       return "#{u.firstname} #{u.lastname}"
10    end
11  end
12
13 end
14
```

course.rb



```
1 class Course < ActiveRecord::Base
2
3   def name
4     return self.title.gsub("-", " ").split.map(&:capitalize).join(' ')
5   end
6
7   def teacher
8     x = User.where(['id = ?', self.teacher_id])
9     if x.blank?
10       "No Teacher."
11     else
12       x.each do |u|
13         return u.name
14       end
15     end
16   end
17
18   def self.hours
19     TimetableEntry.find_by(course_id: self.id).count
20   end
21
22 end
23
```

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

group.rb

The screenshot shows a code editor with the file `group.rb` open. The code defines a class `Group` that inherits from `ActiveRecord::Base`. It has a relationship `has_many :users`. The `leader` method returns the user with the matching `leader_id`, or "No Leader." if none exists. The `users` method returns all users associated with the group. The `lessonstoday` method generates an HTML string for lessons scheduled for the current day.

```
1 class Group < ActiveRecord::Base
2   has_many :users
3
4   def leader
5     x = User.where(['id = ?', self.leader_id])
6     if x.blank?
7       "No Leader."
8     else
9       x.each do |u|
10         return u.name
11       end
12     end
13   end
14
15   def users
16     User.where(['role = ?', self.id])
17   end
18
19   def lessonstoday
20     currnday = Date.today.strftime("%A")
21     TimetableEntry.where(['timetable_id = ? and day = ?', self.timetable_id, currnday]).each do |t|
22       return "<tr><td>#{t.start}</td><td>#{t.course('title')}</td></tr>".html_safe
23     end
24   end
25
26 end
```

login_activity.rb

The screenshot shows a code editor with the file `login_activity.rb` open. It defines a class `LoginActivity` that inherits from `ActiveRecord::Base`. The `user` method returns the user associated with the activity.

```
1 class LoginActivity < ActiveRecord::Base
2
3   def user
4     User.find_by(id: self.person_id)
5   end
6
7 end
```

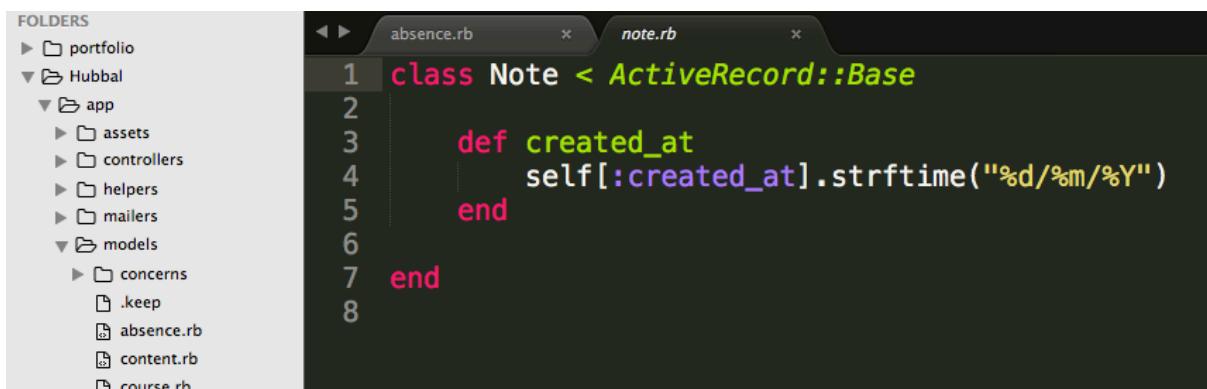
message.rb

The screenshot shows a code editor with the file `message.rb` open. It defines a class `Message` that inherits from `ActiveRecord::Base`. The class includes methods to get the sender's name, recipient's name, and the URL of the sender's profile picture.

```
1 class Message < ActiveRecord::Base
2
3   def sender
4     return User.find_by(id: self.sender_id).name
5   end
6
7   def recipient
8     return User.find_by(id: self.recipient_id).name
9   end
10
11  def sender_picture
12    return User.find_by(id: self.sender_id).avatar.url
13  end
14
15 end
```

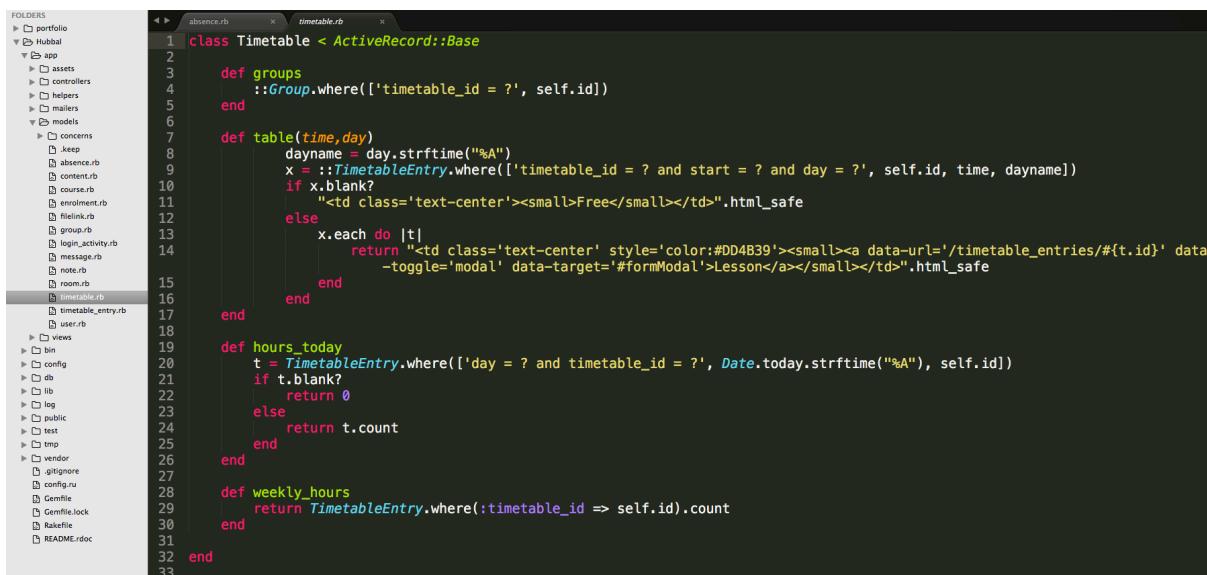
CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

note.rb



```
1 class Note < ActiveRecord::Base
2
3   def created_at
4     self[:created_at].strftime("%d/%m/%Y")
5   end
6
7 end
8
```

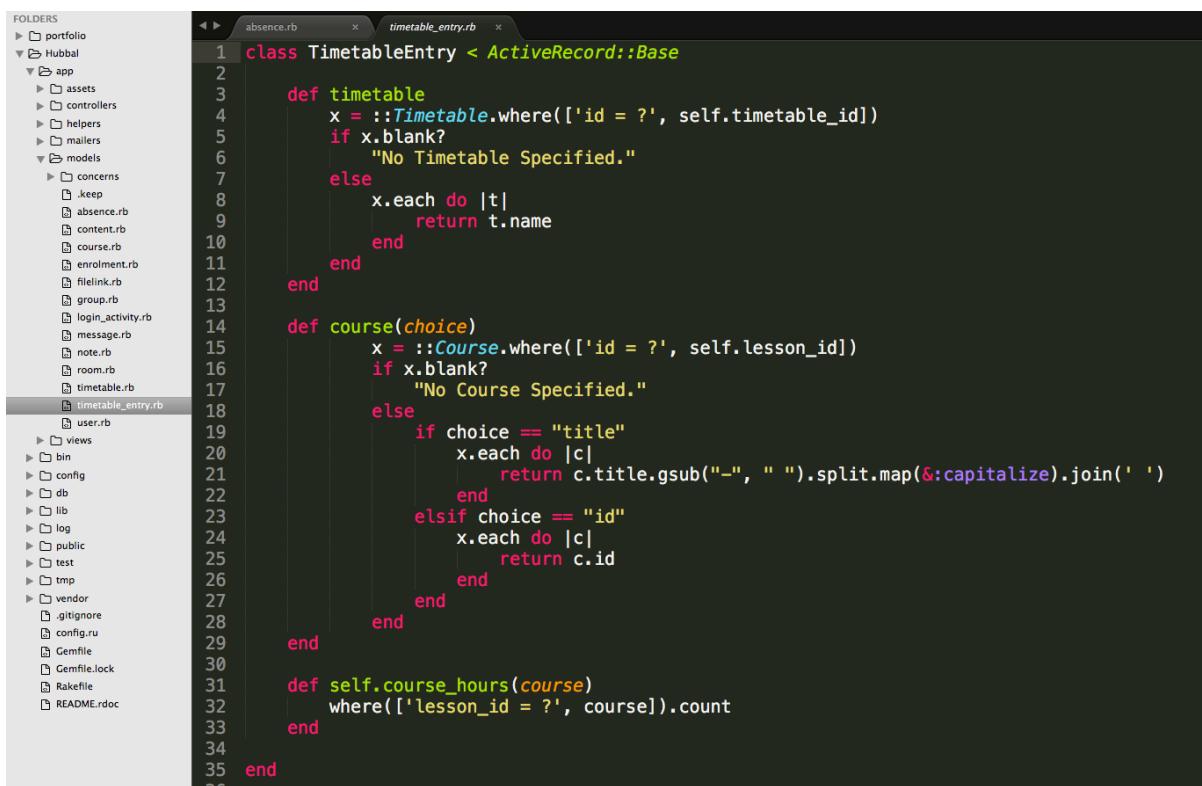
timetable.rb



```
1 class Timetable < ActiveRecord::Base
2
3   def groups
4     ::Group.where(['timetable_id = ?', self.id])
5   end
6
7   def table(time, day)
8     dayname = day.strftime("%A")
9     x = ::TimetableEntry.where(['timetable_id = ? and start = ? and day = ?', self.id, time, dayname])
10    if x.blank?
11      "<td class='text-center'><small>Free</small></td>".html_safe
12    else
13      x.each do |t|
14        return "<td class='text-center' style='color:#DD4B39'><small><a data-url='/timetable_entries/#{t.id}' data-
15          -toggle='modal' data-target='#myModal'>Lesson</a></small></td>".html_safe
16      end
17    end
18  end
19
20  def hours_today
21    t = TimetableEntry.where(['day = ? and timetable_id = ?', Date.today.strftime("%A"), self.id])
22    if t.blank?
23      return 0
24    else
25      return t.count
26    end
27  end
28
29  def weekly_hours
30    return TimetableEntry.where(:timetable_id => self.id).count
31
32 end
33
```

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

timetable_entry.rb



```
FOLDERS
  ▶ portfolio
  ▶ Hubbal
    ▶ app
      ▶ assets
      ▶ controllers
      ▶ helpers
      ▶ mailers
    ▶ models
      ▶ concerns
        □ .keep
      □ absence.rb
      □ content.rb
      □ course.rb
      □ enrolment.rb
      □ filalink.rb
      □ group.rb
      □ login_activity.rb
      □ message.rb
      □ note.rb
      □ room.rb
      □ timetable.rb
      □ timetable_entry.rb
    ▶ views
  ▶ bin
  ▶ config
  ▶ db
  ▶ lib
  ▶ log
  ▶ public
  ▶ test
  ▶ tmp
  ▶ vendor
    □ .gitignore
    □ config.ru
    □ Gemfile
    □ Gemfile.lock
    □ Rakefile
  □ README.rdoc

abscence.rb × timetable_entry.rb ×
1 class TimetableEntry < ActiveRecord::Base
2
3   def timetable
4     x = ::Timetable.where(['id = ?', self.timetable_id])
5     if x.blank?
6       "No Timetable Specified."
7     else
8       x.each do |t|
9         return t.name
10      end
11    end
12  end
13
14  def course(choice)
15    x = ::Course.where(['id = ?', self.lesson_id])
16    if x.blank?
17      "No Course Specified."
18    else
19      if choice == "title"
20        x.each do |c|
21          return c.title.gsub("-", " ").split.map(&:capitalize).join(' ')
22        end
23      elsif choice == "id"
24        x.each do |c|
25          return c.id
26        end
27      end
28    end
29  end
30
31  def self.course_hours(course)
32    where(['lesson_id = ?', course]).count
33  end
34
35 end
36
```

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

user.rb

```
class User < ActiveRecord::Base
  has_secure_password
  has_attached_file :avatar, default_url: "/images/missing.png"
  validates_attachment_content_type :avatar, content_type: /\Aimage\/.*\Z/
  belongs_to :group
  has_many :login_activities

  def group
    Group.where(['id = ?', self.role])
  end

  def timetable_id
    Timetable.where(['user_id = ?', self.id]).each do |t|
      return t.id
    end
  end

  def name
    if self.firstname.blank? and self.lastname.blank?
      return self.username
    else
      "#{self.firstname} #{self.lastname}"
    end
  end

  def status_colour
    if self.status = 0 or self.status.nil?
      "danger"
    elsif self.status = 1
      "info"
    elsif self.status = 2
      "warning"
    elsif self.status = 3
      "default"
    elsif self.status = 4
      "success"
    end
  end

  def level
    self.group.each do |x|
      return x.name
    end
  end

  def status_feedback
    if self.status = 0 or self.status.nil?
      "<i class='fa fa-circle text-danger'></i> Late".html_safe
    elsif self.status = 1
      "<i class='fa fa-circle text-info'></i> Sickness Absence".html_safe
    elsif self.status = 2
      "<i class='fa fa-circle text-warning'></i> Holiday Absence".html_safe
    elsif self.status = 3
      "<i class='fa fa-circle text-default'></i> Free Period".html_safe
    elsif self.status = 4
      "<i class='fa fa-circle text-success'></i> On Schedule".html_safe
    end
  end

  def logs
    LoginActivity.where(['person_id = ?', self.id])
  end

  def employer
    return Group.find_by(id: self.role).leader
  end
end
```

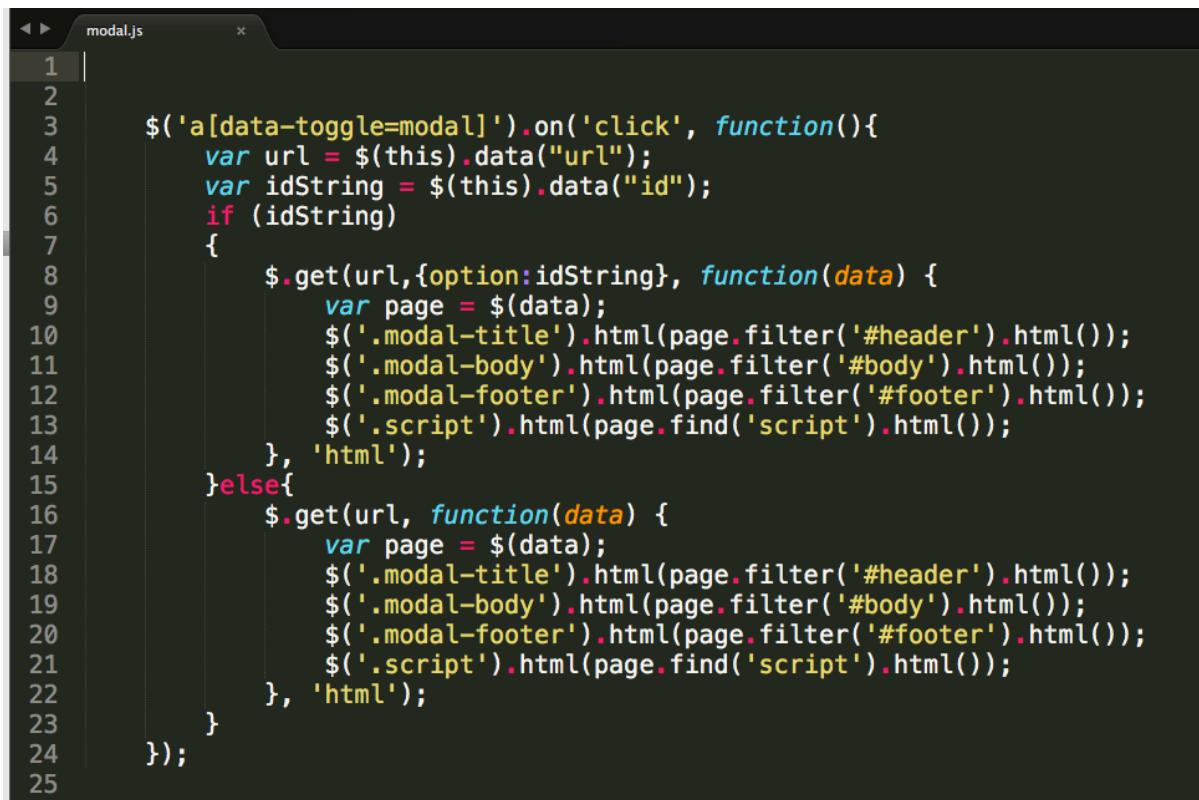
Appendix GG: User Controller

```

users_controller.rb ×
1 class UsersController < ApplicationController
2   before_action :set_user, only: [:show, :edit, :update, :destroy]
3
4   # GET /users
5   # GET /users.json
6   def index
7     @users = User.all
8   end
9
10  # GET /users/1
11  # GET /users/1.json
12  def show
13    end
14
15  # GET /users/new
16  def new
17    @user = User.new
18    render :layout => "modal"
19  end
20
21  # GET /users/1/edit
22  def edit
23    render :layout => "modal"
24  end
25
26  # POST /users
27  # POST /users.json
28  def create
29    params[:user][:username] = "#{rand(99).to_s}#{params[:user][:firstname].split("//).first}#{params[:user][:lastname].strip}#{rand(99).to_s}"
30    @user = User.new(user_params)
31
32    respond_to do |format|
33      if @user.save
34        format.html { redirect_to @user, notice: 'User was successfully created.' }
35        format.json { render :show, status: :created, location: @user }
36      else
37        format.html { render :new }
38        format.json { render json: @user.errors, status: :unprocessable_entity }
39      end
40    end
41    @new_user = User.find_by(username: @user.username)
42    Timetable.create(:user_id => @new_user.id)
43  end
44
45  # PATCH/PUT /users/1
46  # PATCH/PUT /users/1.json
47  def update
48    respond_to do |format|
49      if @user.update(user_params)
50        format.html { redirect_to @user, notice: 'User was successfully updated.' }
51        format.json { render :show, status: :ok, location: @user }
52      else
53        format.html { render :edit }
54        format.json { render json: @user.errors, status: :unprocessable_entity }
55      end
56    end
57  end
58
59  # DELETE /users/1
60  # DELETE /users/1.json
61  def destroy
62    @user.destroy
63    respond_to do |format|
64      format.html { redirect_to users_url, notice: 'User was successfully destroyed.' }
65      format.json { head :no_content }
66    end
67  end
68
69  private
70  # Use callbacks to share common setup or constraints between actions.
71  def set_user
72    @user = User.find(params[:id])
73  end
74
75  # Never trust parameters from the scary internet, only allow the white list through.
76  def user_params
77    params.require(:user).permit(:firstname, :lastname, :username, :password, :email, :phone, :role, :room, :avatar)
78  end
79 end
80

```

Appendix HH: Modal.js Source Code



A screenshot of a code editor window titled "modal.js". The code is written in JavaScript and uses jQuery syntax. It handles a click event on an anchor tag with a "data-toggle=modal" attribute. It retrieves a URL from the "data-url" attribute and an ID string from the "data-id" attribute. If an ID string is present, it performs a GET request to the URL with an option parameter set to the ID string. The response is then parsed into a page object, which is filtered to remove header, body, and footer elements, and its script element is also removed. The resulting HTML is then assigned to the modal's title, body, and footer. If no ID string is present, it simply performs a GET request to the URL and assigns the entire response to the modal's title, body, and footer. The code ends with a closing parenthesis and a semicolon.

```
1
2
3     $('a[data-toggle=modal]').on('click', function(){
4         var url = $(this).data("url");
5         var idString = $(this).data("id");
6         if (idString)
7         {
8             $.get(url,{option:idString}, function(data) {
9                 var page = $(data);
10                $('.modal-title').html(page.filter('#header').html());
11                $('.modal-body').html(page.filter('#body').html());
12                $('.modal-footer').html(page.filter('#footer').html());
13                $('.script').html(page.find('script').html());
14            }, 'html');
15        }else{
16            $.get(url, function(data) {
17                var page = $(data);
18                $('.modal-title').html(page.filter('#header').html());
19                $('.modal-body').html(page.filter('#body').html());
20                $('.modal-footer').html(page.filter('#footer').html());
21                $('.script').html(page.find('script').html());
22            }, 'html');
23        }
24    });
25 }
```

Appendix II: Frontend Source Code

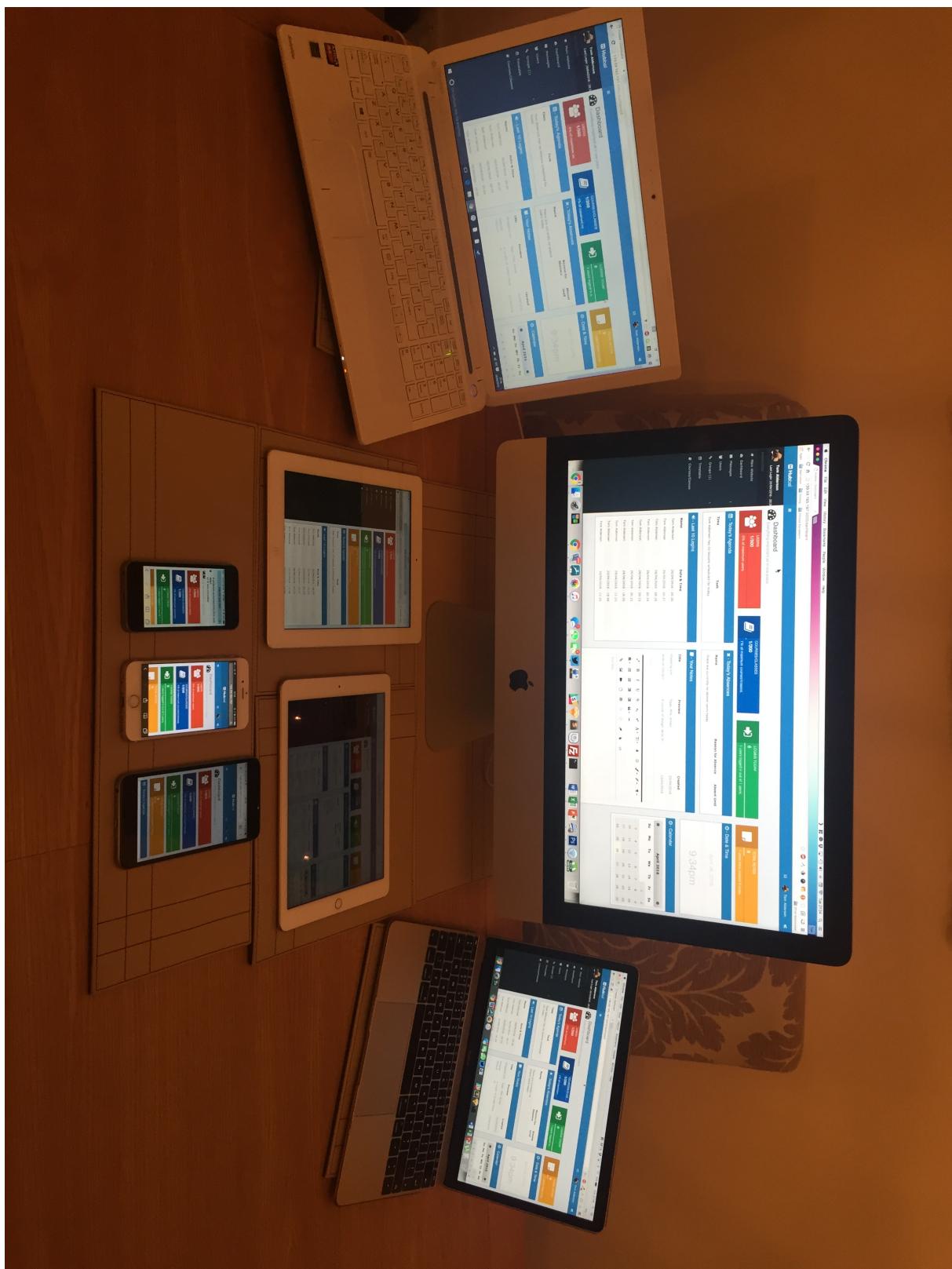
```

frontend.html.erb

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Silverleaf College &quot; Developing the future together.</title>
6     <meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" name="viewport">
7     <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css" rel="stylesheet" type="text/css" />
8     <link href="https://code.ionicframework.com/ionicframework/2.0.1/css/ionicons.min.css" rel="stylesheet" type="text/css" />
9     <link href="https://fonts.googleapis.com/css?family=Open+Sans:400,300,300italic,400italic,500,500italic,700,700italic" rel="stylesheet" type="text/css" />
10    <!-- stylesheets -->
11    <!-- HTML5 Shim and Respond.js support of HTML5 elements and media queries -->
12    <!-- WARNING: Respond.js doesn't work if you view the page via file://>
13    <!-- If lt IE 9 -->
14    <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.2/html5shiv.js"></script>
15    <script src="https://oss.maxcdn.com/libs/respond.min.js"></script>
16  <!--endif-->
17  <!-- csrf_meta_tags -->
18 </head>
19 <body class="frontend">
20   <div class="navigation-bar">
21     <div class="app"></div>
22     <div class="navigation-bar__container">
23       <div class="navigation-bar__item"><a href="/">Home</a></div>
24       <div class="navigation-bar__item"><a href="/about">About</a></div>
25       <div class="navigation-bar__item"><a href="/history">History</a></div>
26       <div class="navigation-bar__item navigation-bar__item--dropdown">
27         <div class="navigation-bar__dropdown-label">Courses</div>
28         <div class="navigation-bar__dropdown-container">
29           <div class="course" each="c" ><a href="/course/<%= c.title %>">&quot; <%= c.name %></a></div>
30         </div>
31       </div>
32     </div>
33   </div>
34   <div class="navigation-bar__item"><a href="/location">Location</a></div>
35   <div class="navigation-bar__item"><a href="/support">Support</a></div>
36 </div>
37 </div>
38 <div class="banner">
39   <div class="banner__content">
40     <div class="banner__title">Silverleaf College</div>
41     <div class="banner__subtitle">Developing the future together.</div>
42   </div>
43 </div>
44 <div class="mid-logo">
45   
46 </div>
47 <div class="container">
48   <div class="faq">
49     <div class="row">
50       <div class="col-md-4">
51         <div class="faq_box">
52           <div class="faq_title">Why Silverleaf College?</div>
53           <div class="faq_content">Silverleaf College is known for it's prestigious results from Ofsted and we pride ourselves on our reputation, in the past five years we have continued to deliver forever-improving results in GCSE exams.</div>
54         </div>
55       </div>
56       <div class="col-md-4">
57         <div class="faq_box">
58           <div class="faq_title">How do I enrol my child?</div>
59             <div class="faq_content">Enrolment here at Silverleaf College is a simple process, requiring a simple phone call to request a prospectus and application form, from there all you need to do is fill in and return the completed forms.</div>
60           </div>
61         </div>
62       <div class="col-md-4">
63         <div class="faq_box">
64           <div class="faq_title">What courses are available?</div>
65           <div class="faq_content">You will be able to find all of our available courses to enrol on in the 'Courses' section located in the navigation, alternatively, you can click <a href="#">here</a>.</div>
66         </div>
67       </div>
68     </div>
69   </div>
70   <section class="content">
71     <div yield>
72   </section>
73 </div>
74 <div class="footer frontend-footer">
75   Website developed for testing purposes of the Hubbal application. Silverleaf College is pure fictional and is in no way related to the Hubbal application besides testing.
76 </div>
77 </div>
78 </html>

```

Appendix JJ: Evidence of Device Testing



Appendix KK: Evidence of Timetabling Options

Personal Timetable View

The screenshot shows the Hubbal web application interface. At the top right, there is a user profile for 'Tom Alderson' with a last login timestamp of '26/04/2016 - 20:32'. Below the profile is a navigation bar with links: 'Main Website', 'Dashboard', 'Messages', and 'Users'. On the left, there is a sidebar with icons for 'Groups (1)', 'Timetable', 'Courses/Classes', and a collapsed section. The main content area is titled '- Timetable' and displays a weekly schedule grid. The grid has columns for 'Time' (8:00 to 17:00) and days of the week (Monday through Thursday). Each cell indicates if the time slot is 'Free' or 'Booked'. The 'Timetable' link in the sidebar is highlighted.

Time	Monday	Tuesday	Wednesday	Thursday
08:00	Free	Free	Free	Free
09:00	Free	Free	Free	Free
10:00	Free	Free	Free	Free
11:00	Free	Free	Free	Free
12:00	Free	Free	Free	Free
13:00	Free	Free	Free	Free
14:00	Free	Free	Free	Free
15:00	Free	Free	Free	Free
16:00	Free	Free	Free	Free
17:00	Free	Free	Free	Free

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Tom Alderson

+ New Event

Thursday	Friday	Saturday	Sunday
Free	Free	Lesson	Free
Free	Free	Lesson	Free
Free	Free	Free	Free
Free	Free	Free	Free
Free	Free	Free	Free
Free	Free	Free	Free
Free	Free	Free	Free
Free	Free	Free	Free
Free	Free	Free	Free
Free	Free	Free	Free
Free	Free	Free	Free
Free	Free	Free	Free

Developed by Thomas Alderson

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Event View

The screenshot shows a web interface for managing events. At the top right, there's a user profile for 'Tom Alderson' with a last login timestamp of '26/04/2016 - 20:32'. Below the profile is a navigation bar with links: 'Main Website', 'Dashboard', 'Messages', and 'Users'. On the left, there's a sidebar with icons for 'Groups (1)', 'Timetable', and 'Courses/Classes'. The main content area displays an event titled 'Website Development' with details: 'Day: Saturday', 'Start Time: 08:00 AM', and 'Room: W2'. A red 'Delete' button is visible. The central part of the screen shows a 'Timetable' grid for Monday through Thursday. The grid has time slots from 08:00 to 17:00. Most slots are marked as 'Free', except for a few specific times like 11:00, 12:00, and 13:00 which are listed as 'Free' under the 'Tuesday' column.

Time	Monday	Tuesday	Wednesday	Thursday
08:00	Free	Free	Free	Free
09:00	Free	Free	Free	Free
10:00	Free	Free	Free	Free
11:00	Free	Free	Free	Free
12:00	Free	Free	Free	Free
13:00	Free	Free	Free	Free
14:00	Free	Free	Free	Free
15:00	Free	Free	Free	Free
16:00	Free	Free	Free	Free
17:00	Free	Free	Free	Free

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

The screenshot displays a web-based application for managing event attendance. The main view is a weekly calendar grid. The days of the week are labeled as follows:

- Thursday:** Free
- Friday:** Free
- Saturday:** Lesson
- Sunday:** Free

Below the calendar, there is a button labeled "+ New Event". A modal window is currently overlaid on the page, containing the following elements:

- A "Close" button in the top left corner.
- A "Submit" button in the top right corner.
- A text input field in the center.
- A small "X" icon in the top right corner of the modal window.

In the top right corner of the main interface, there is a user profile section for "Tom Alderson" which includes a photo, a mail icon, and a gear icon.

Developed by Thomas Alderson

User Profile Scheduling

The screenshot shows a web interface for managing user profiles and scheduling. At the top, there's a header bar with the title "User Profile Scheduling". Below this is a navigation sidebar on the left containing links for "Main Website", "Dashboard", "Messages", "Users", "Groups (1)", "Timetable", and "Courses/Classes". The main content area is divided into several sections:

- Profile Section:** Displays a profile picture of a man with blonde hair, the name "Tom Alderson", and the title "Administrator". It also shows the last login date: "Last Login: 26/04/2016 - 20:34".
- Information Section:** Titled "i - Basic Information", it contains a blue header and a white body.
- Timetable Section:** Titled "calendar - Timetable Today", it shows a message: "Tom Alderson has no lessons scheduled for today."
- Login Activity Section:** Titled "key - Login Activity", it lists logins by time and IP address.

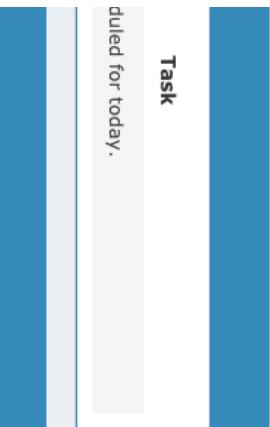
Time	IP Address
08/04/2016 - 14:46	127.0.0.1
08/04/2016 - 15:27	127.0.0.1
10/04/2016 - 11:25	127.0.0.1
22/04/2016 - 19:56	127.0.0.1
23/04/2016 - 11:25	127.0.0.1
24/04/2016 - 16:20	94.192.114.121
26/04/2016 - 20:21	94.192.114.121
26/04/2016 - 20:23	94.192.114.121
26/04/2016 - 20:24	94.192.114.121
26/04/2016 - 20:25	94.192.114.121
26/04/2016 - 20:27	94.192.114.121
26/04/2016 - 20:30	94.192.114.121
26/04/2016 - 20:32	94.192.114.121

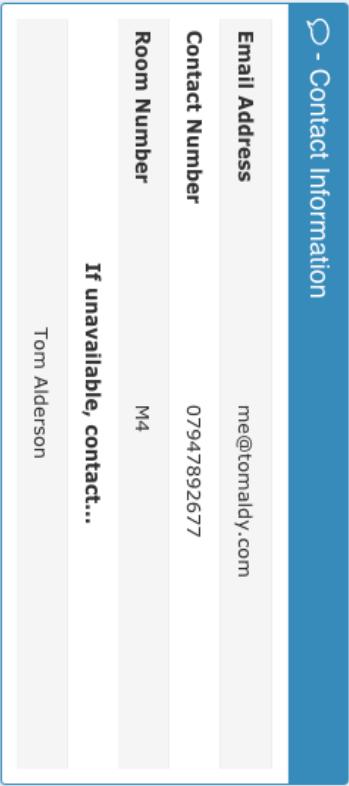
CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

Q4 197 114 171

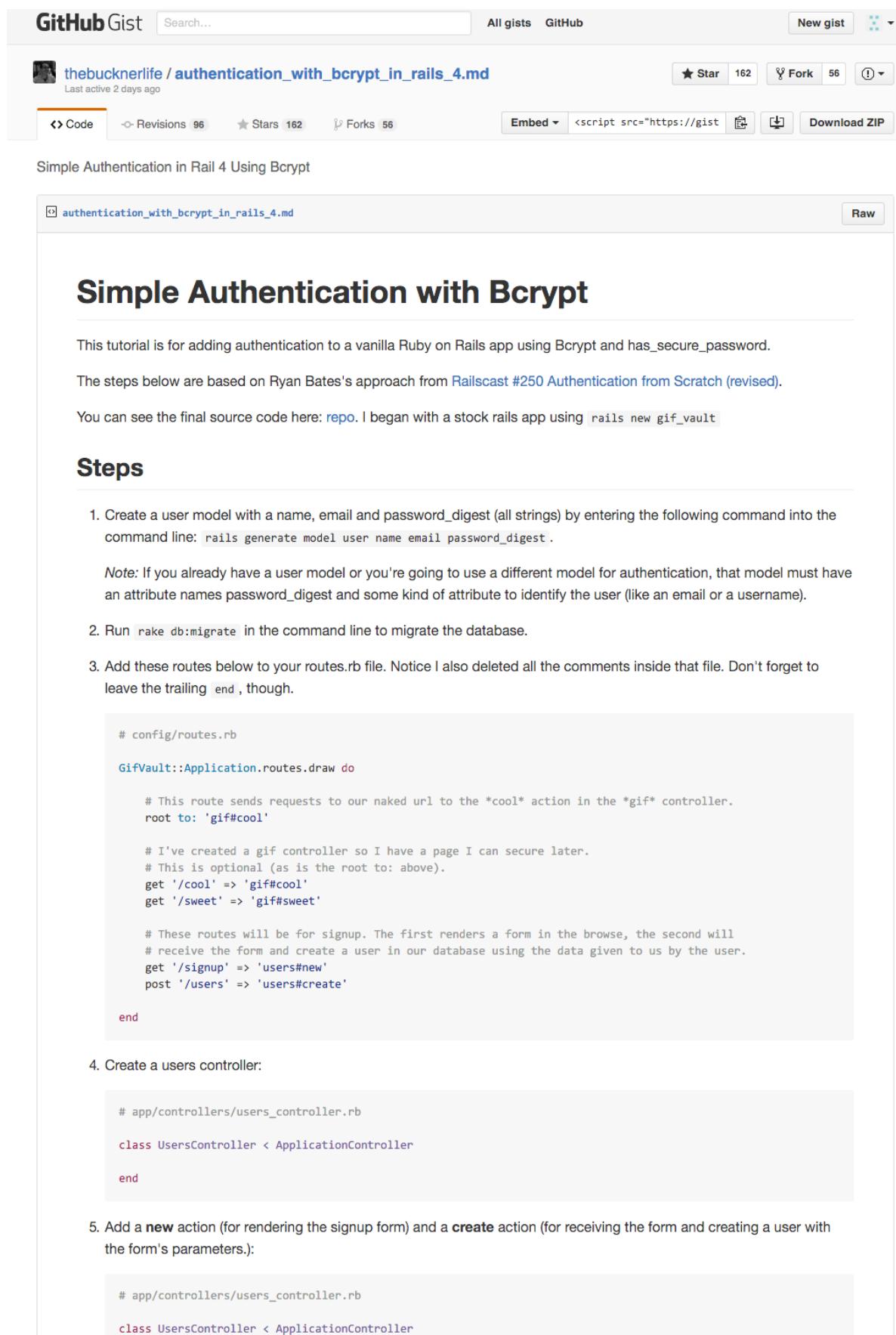
 Tom Alderson 

 + Send Message  Edit User


Task
duled for today.


Email Address me@tomaldy.com
Contact Number 07947892677
Room Number M4
If unavailable, contact...
Tom Alderson

Appendix LL: Article on 'password_digest'



The screenshot shows a GitHub Gist page for a file named 'authentication_with_bcrypt_in_rails_4.md'. The page title is 'Simple Authentication with Bcrypt'. The content discusses adding authentication to a Ruby on Rails app using Bcrypt and has_secure_password. It includes steps for creating a user model, running migrations, and adding routes. It also shows the source code for config/routes.rb and app/controllers/users_controller.rb.

Simple Authentication with Bcrypt

This tutorial is for adding authentication to a vanilla Ruby on Rails app using Bcrypt and has_secure_password.

The steps below are based on Ryan Bates's approach from Railscast #250 Authentication from Scratch (revised).

You can see the final source code here: [repo](#). I began with a stock rails app using `rails new gif_vault`

Steps

1. Create a user model with a name, email and password_digest (all strings) by entering the following command into the command line: `rails generate model user name email password_digest`.
Note: If you already have a user model or you're going to use a different model for authentication, that model must have an attribute names password_digest and some kind of attribute to identify the user (like an email or a username).
2. Run `rake db:migrate` in the command line to migrate the database.
3. Add these routes below to your routes.rb file. Notice I also deleted all the comments inside that file. Don't forget to leave the trailing `end`, though.

```
# config/routes.rb

GifVault::Application.routes.draw do

  # This route sends requests to our naked url to the *cool* action in the *gif* controller.
  root to: 'gif#cool'

  # I've created a gif controller so I have a page I can secure later.
  # This is optional (as is the root to: above).
  get '/cool' => 'gif#cool'
  get '/sweet' => 'gif#sweet'

  # These routes will be for signup. The first renders a form in the browse, the second will
  # receive the form and create a user in our database using the data given to us by the user.
  get '/signup' => 'users#new'
  post '/users' => 'users#create'

end
```

4. Create a users controller:

```
# app/controllers/users_controller.rb

class UsersController < ApplicationController

end
```

5. Add a **new** action (for rendering the signup form) and a **create** action (for receiving the form and creating a user with the form's parameters.):

```
# app/controllers/users_controller.rb

class UsersController < ApplicationController
```

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

```
    . . .
end
```

5. Add a **new** action (for rendering the signup form) and a **create** action (for receiving the form and creating a user with the form's parameters.):

```
# app/controllers/users_controller.rb

class UsersController < ApplicationController

  def new
  end

  def create
  end

end
```

6. Now create the view file where we put the signup form.

```
<!-- app/views/users/new.html.erb -->

<h1>Signup!</h1>

<%= form_for :user, url: '/users' do |f| %>

  Name: <%= f.text_field :name %>
  Email: <%= f.text_field :email %>
  Password: <%= f.password_field :password %>
  Password Confirmation: <%= f.password_field :password_confirmation %>
  <%= f.submit "Submit" %>

<% end %>
```

A note on Rail's conventions: This view file is for the **new** action of the **users controller**. As a result, we save the file here: `/app/views/users/new.html.erb`. The file is called **new.html.erb** and it is saved inside the views folder, in a folder we created called **users**.

That's the convention: view files are inside a folder with the same name as the controller and are named for the action they render.

7. Add logic to **create** action and add the private `user_params` method to sanitize the input from the form (this is a new Rails 4 thing and it's required). You might need to adjust the parameters inside the `.permit()` method based on how you setup your User model.

```
class UsersController < ApplicationController

  def new
  end

  def create
    user = User.new(user_params)
    if user.save
      session[:user_id] = user.id
      redirect_to '/'
    else
      redirect_to '/signup'
    end
  end

  private

  def user_params
    params.require(:user).permit(:name, :email, :password, :password_confirmation)
  end
end
```

8. Go to your Gemfile and uncomment the 'bcrypt' gem. We need bcrypt to securely store passwords in our database.

```
source 'https://rubygems.org'

# Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
gem 'rails', '4.0.4'
```

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

```
gem 'rails', '4.0.4'

# Use sqlite3 as the database for Active Record
gem 'sqlite3'

...

# Use ActiveModel has_secure_password
gem 'bcrypt', '~> 3.1.7'

...
```

9. Go to the User model file and add `has_secure_password`. This is the line of code that gives our User model authentication methods via bcrypt.

```
# app/models/user.rb

class User < ActiveRecord::Base

  has_secure_password

end
```

10. Run `bundle install` from the terminal then restart your rails server.

Note: Windows users might have issues with bcrypt. If so, copy the error into Google and look for answers on Stack Overflow. There is documentation online for how to fix Windows so the bcrypt works.

11. Create a sessions controller. This is where we create (aka login) and destroy (aka logout) sessions.

```
# app/controllers/sessions_controller.rb

class SessionsController < ApplicationController

  def new
  end

  def create
  end

  def destroy
  end

end
```

12. Create a form for user's to login with.

```
<!-- app/views/sessions/new.html.erb -->

<h1>Login</h1>

<%= form_tag '/login' do %>

  Email: <%= text_field_tag :email %>
  Password: <%= password_field_tag :password %>
  <%= submit_tag "Submit" %>

<% end %>
```

13. Update your routes file to include new routes for the sessions controller.

```
GifVault::Application.routes.draw do

  root to: 'gif#cool'

  # these routes are for showing users a login form, logging them in, and logging them out.
  get '/login' => 'sessions#new'
  post '/login' => 'sessions#create'
  get '/logout' => 'sessions#destroy'

  get '/signup' => 'users#new'
  post '/users' => 'users#create'

end
```

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

14. Update the sessions_controller with the logic to log users in and out.

```
# app/controllers/sessions_controller.rb

def create
  user = User.find_by_email(params[:email])
  # If the user exists AND the password entered is correct.
  if user && user.authenticate(params[:password])
    # Save the user id inside the browser cookie. This is how we keep the user
    # logged in when they navigate around our website.
    session[:user_id] = user.id
    redirect_to '/'
  else
    # If user's login doesn't work, send them back to the login form.
    redirect_to '/login'
  end
end

def destroy
  session[:user_id] = nil
  redirect_to '/login'
end
```

15. Update the application controller with new methods to look up the user, if they're logged in, and save their user object to a variable called @current_user. The helper_method line below current_user allows us to use @current_user in our view files. Authorize is for sending someone to the login page if they aren't logged in - this is how we keep certain pages our site secure... user's have to login before seeing them.

```
# app/controllers/application_controller.rb

class ApplicationController < ActionController::Base
  # Prevent CSRF attacks by raising an exception.
  # For APIs, you may want to use :null_session instead.
  protect_from_forgery with: :exception

  def current_user
    @current_user ||= User.find(session[:user_id]) if session[:user_id]
  end
  helper_method :current_user

  def authorize
    redirect_to '/login' unless current_user
  end
end
```

16. Add a before_filter to any controller that you want to secure. This will force user's to login before they can see the actions in this controller. I've created a gif controller below which I'm going to secure. The routes for this controller were added to the routes.rb in the beginning of this tutorial.

```
# app/controllers/gif_controller.rb

class GifController < ApplicationController
  before_filter :authorize

  def cool
  end

  def free
  end
end
```

17. You can update your application layout file to show the user's name if they're logged in and some contextual links.

```
<!-- app/views/layout/application.html.erb -->

<!DOCTYPE html>
<html>
<head>
  <title>GifVault</title>
```

CAN A WEB INTERFACE IMPROVE THE QUALITY, RELIABILITY AND ACCESSIBILITY OF AN ATTENDANCE TRACKING SYSTEM FOR AN ORGANISATION?

```
----->
<%= stylesheet_link_tag    "application", media: "all", "data-turbolinks-track" => true %>
<%= javascript_include_tag "application", "data-turbolinks-track" => true %>
<%= csrf_meta_tags %>
</head>
<body>

# added these lines.
<% if current_user %>
  Signed in as <%= current_user.name %> | <%= link_to "Logout", '/logout' %>
<% else %>
  <%= link_to 'Login', '/login' %> | <%= link_to 'Signup', '/signup' %>
<% end %>

<%= yield %>

</body>
</html>
```

Things Missing

- Adding flash messages would be simple and provide feedback to the user if things go wrong.

All done! Feel free to fork and update this. Reach me at @thebucknerlife on Twitter.

End of Report.

Can a Web Interface improve the quality, reliability and accessibility of an Attendance Tracking system for an Organisation?

Written & Produced by Thomas Alderson