RMIT - OUA
Building IT Systems CPT111

Assignment 3

Study Period 1, 2021

Prepared by Group 14

Daniel Melfa	S3866981
Patrick Jenner	S3831040
Moustafa Al-Meahi	S3887033
Tyson Carroll	S3660241
Megha Patel	S3858020

This document has been prepared as a deliverable for the RMIT OUA Building IT Systems (CPT111) course Assignment 3 Stage 1 for the study Period 1 2021.

Table of Contents

1. Project Background

1.1. Project Description

The application that our group has chosen to work on is a web application (website) named “Boomerang”. The website is a platform where users can exchange diverse tools and hardware that they own in the form of a lease and/or rental agreement. The platform allows users to create their profiles and list items that they are willing to lease on an agreed period.

After considerable deliberation, we validated that there was no application on the market that allowed users to lease or rent hardware tools from each other, this was a slice of market almost entirely untouched.

During the initial ideation and design phases of the project the team agreed on what the minimum viable features would be, in additional a few extended features were outlined and deemed to be “nice to have(s)”

At a minimum, the Boomerang web application would allow users to search for items based on a variety of methods, including category and sub-categories, product name and other attributes. The goal of the group was to make this experience as seamless as possible to the user and provide the user a rich experience that included photos of the items, reviews from previous users, and provide hints & tips on how to use the tools. In addition, we decided that the ability to filter matching results based on post code would further enhance the user experience.

When using the system for a particular “transaction”, a user would be tagged as either a “lessor” (the person who is leasing the item to a third party) or a “renter” (the person who is renting the item from a third party). There are no limitations imposed to “what type” of user a person can be, all registered users are able to lease their tools as well as rent someone else’s tools.

The process of listing items on the website is designed to be intuitive, users can add photos of their items, specify prices, add information about the tools, etc.

Going beyond the minimum viable product (minimum viable features), the group agreed that a user feedback and review system would be highly beneficial to all users. Users of the system are subjected to a user rating system where all parties involved in a transaction can provide feedback about each other. To illustrate this, imagine that a “lessor” can provide feedback about a “renter” to share information about the item leased and disclose whether the “renter” looked after the item well or not. On the same token the “renter” can disclose feedback about the tool to tell other users if the rented item was in good condition and if it worked as advertised.

To further improve the usability of the system, a notifications system was also part of the “extended features”, this feature was designed to allow users to ask questions about specific items and allow the “lessors” to respond to the potential renters’ inquiries.

Another advanced feature would be to allow users to make payments directly via the website, this would avoid users having to pay for their transactions in cash, thus improving the reliability of the system and avoid potential misunderstandings in their dealings.

In the next section below each member of the group motivation, aims and goals are highlighted.

1.2. Project Team Description

Patrick Jenner

Student Number: S3831040
Student Email Address: S3831040@student.rmit.edu.au

Motivation:
I’m a keen learner who was motivated to start a project which involved me learning new skills and tools to create something new. I’m forever trying to expand my horizons and see how far I can go. During this project I was keen to learn about web development and the processes involved with getting building a platform from idea to reality. This project has allowed me to gain insights into business operations as well as more technical web development knowledge.

Role: Frontend Web Developer

Personal Background:
I have worked previously on and off in the IT industry doing mostly helpdesk, technical support, hardware maintenance and sales roles. I am trying to focus my career goals and discover and path that can direct my efforts which I am most suited to. I have a technical understand and can get an idea how something is working from a larger perspective.

Project Aim:
During this project I hoped to gain more knowledge on web development using Django, python and the process involved with integrating the front and back-end processes.

Project Goals:

1. Learn what's involved with getting an idea from concept to a reality
2. Learn Django and how to integrate a database with a frontend functional website
3. Learn the basic ins and outs of HTML and CSS and getting a modern website up and running

Daniel Melfa

Student Number: S3866981
Student Email Address: S3866981@student.rmit.edu.au

Motivation:

My motivation mainly revolved around learning collaboration techniques such as working in an agile mode, learning about "Kanban-style" project management tools (i.e., Trello), improving our skills on how to identify and solve problems commonly associated with the implementation of Information Technology projects. In addition, I was also interested in learning about presentation skills, and perhaps learning something new about software tools that I have not had the chance to use before.

Role:

Team Lead, Database Administrator

Personal Background:

I have been working in the IT industry since the year 2000, my background has mostly been working as a Pre-Sales Technical Consultant for various IT companies. Common duties that I perform at work are to run software/solution demonstrations, run technical workshops to explain to potential clients how the software works, document requirements, and oversee implementations.

Project Aim:

The aim of the project "Boomerang" was to help users lease/rent tools and hardware in a peer-to-peer manner, facilitated by a centralised web application.

Project Goals:

1. We identified a use case that was not addressed by any other tool on the market
2. The group created a high-fidelity prototype of how the application would work and what it would like
3. We designed the logic behind various functions of the application.

Megha Patel

Student Number: S3858020
Student Email Address: S3858020@student.rmit.edu.au
Roles: Project Manager, Business Analyst

Motivation:

My motivation in this project is twofold. Firstly, I wanted to gain experience with Django or similar devices and broaden my knowledge about software and hardware integration. Joining a group of other enthusiasts was an excellent way for me to begin this journey. I was looking forwards to learning about different programming languages and databases and integrating them into Django.

Background:

I am currently working as a CSR Supervisor. I regularly work on projects and in Agile environment and understand the process and how a team needs to work together to keep a deadline. I have no professional experience in IT. Having a personal interest in IT and computer science concepts, I am pursuing formal education in this field.

Despite my lack of experience, I am passionate about programming and development. I especially enjoy the puzzle-like frameworks that go into designing a program. I really love networking, as well as all the technical information that fall under the larger umbrella of networking. I have developed some skills through self-study.

Project Aim:

Create a functional website that allows users to list their products for hire using Django.

Project Goals:

1. Learned a new programming language and framework.
2. Coordinate working on a project with a team working entirely remotely.
3. Expanded my knowledge of the full software development lifecycle.

Moustafa Al-Meahi

Student Number: S3887033
Student Email Address: S3887033@student.rmit.edu.au
Roles: General Programmer

Motivation:

I'm roused to Endeavor to finish our Web Application project so I can gain proficiency for certain new abilities. The new mastery that I am wanting to learn are Python and HTML coding, how to utilize the Django Framework and figure out how to do a touch of information base work.

Background:
I have been learning Python and other programming languages as a hobby in my free time during my time in High school. I have always been fascinated on pursuing a career in the Information Technology industry, especially Cybersecurity.

Project Aim:
To develop and publish a functioning website to provide a platform for our users.

Project Goals:

1. Learned how to use the Django Framework
2. Developed team skills and using third party tools to excel our team productivity
3. Learnt how to program with SQL

Tyson Carroll

Student Number: s3660241
Student Email Address: s3660241@student.rmit.edu.au
Roles: Project Manager, Back-End Django Developer.

Motivation:
I am motivated to attempt to complete our Web Application project so I can learn some new skills. The new expertise that I am hoping to learn are Python and HTML coding, how to use the Django Framework and learn how to do a bit of database work.

Background:
I grew a passion for I.T from my love of video games from age, building computers and helping friends with minor PC problems. I have always wanted to develop a game and I hopefully will do in future. I am also interested in developing skills in Cybersecurity later in life, to hopefully get a job in the field of Cyber Security or I am also interested in a job in Game Development.

Project Aim:
To create a functional website in the Django Framework.

Project Goals:
Three goals that I achieved in order to compete my project aim were:

1. Learn the basics of Python and HTML coding
2. Start building the initial stages of the website with the Django framework
3. Allow users to create a login which they can currently use to sign in to the website (This data is stored in an SQLite database)

2. Project Process

2.1. Description

The Group 14 (codenamed "Alpha Team") was a team that was formed late, probably one of the last ones that was formed. From the beginning we were a week behind the schedule in terms of having selected what stream of technologies we would follow (Web applications, mobile, games, Discord Bots, etc.)

One of the first challenges we faced in the initial few weeks of the course was that the team was not formed based on its member mutual interests (like most other teams), but rather it was a team that was formed out of students that had not yet joined a group. The situation we faced was that some students wanted to start a project for a game, others wanted to start a project for Discord Bot, and other a web application. The team spent almost 2 weeks in reflecting what project stream would provide the team members a good mix of skills and also, we wanted to follow a stream that had plenty of "how-to" tutorials available on the internet. The team finally agreed that we would pursue the "Web application" stream.

The next phase of our work (weeks 3 to 5) was to decide on what exactly our web application would do, the group "brainstormed" for a few days on what kinds of applications (either web or mobile) each group member was familiar with, we discussed about various popular applications such as Uber, Airbnb, eBay, Amazon, Go Get, Air Tasker and others. The aim of the discussions was to identify what other sort of functionality would be of value to users and that it had not yet been developed by any other software companies. **As described in the "Project Background" section of this document**, after many meetings and conversations we decided on the peer-to-peer Tools and Hardware lease/rentals platform named "Boomerang".

In the weeks 4 and 5 the team worked closely together in the creation of the Project Proposal, the document detailing the initial proposed features was submitted mid-week 5, on the 31st of March 2021.

After having decided on what the technology stream would be and what the application would do, and having submitted the initial project proposal, the next step would be to decide on the roles of each one in the team. The initial roles that we defined were:

- Daniel Melfa – Solution Architect, Team Leader
- Megha Patel – Project Manager, Business Analyst
- Moustafa Al-Meahi – General Programmer
- Tyson Carroll – Project Manager, Back-End Django Developer.
- Patrick Jenner – Front End Designer

As the design phase of the project started (weeks 6 to 8), it soon became clear to the team that the previously defined roles were not closely followed, in the multiple workshops we had online everybody contributed in many ways, contributions were related to the functionality of each feature, what each interface would look like, what the user experience should be and more. Most of this design work was done using the prototyping platform FIGMA.

Our team met on an average of 4 to 6 hours per week (2 weekly meetings), during these meetings everyone provided input on what the minimum viable features (MVF) would be, as well as the extended viable features (EVF). During these two weeks (6 to 8) we agreed how the work would be split amongst each team member.

The work was split as follows. Refer to Table 1:

Table1

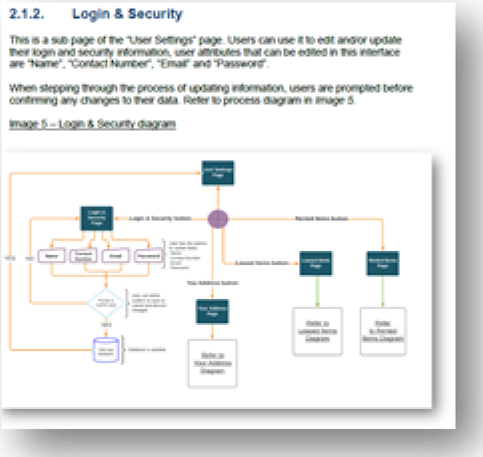
Daniel Melfa	User Profile sections “Login & Security” and “My Leased Items”. In addition, as the Team Lead, I had the role of integrate all functions together.
Megha Patel	User Profile sections “My Address” and “My Rented Items”
Moustafa Al-Meahi	Add Items Functionality
Tyson Carroll	Notifications System
Patrick Jenner	Search Functionality and Product Listing

The tasks associated with the split of work included:

1) Design the logic and steps respective to that feature.
2) Create a high-fidelity prototype in FIGMA of the feature
3) Diagram and document the rules and logic of the feature for the Assignment 2
4) Create a step-by-step guide for the feature for the Assignment 2

Examples of the content that was created, refer to image 1 and 2 below:

Image 1



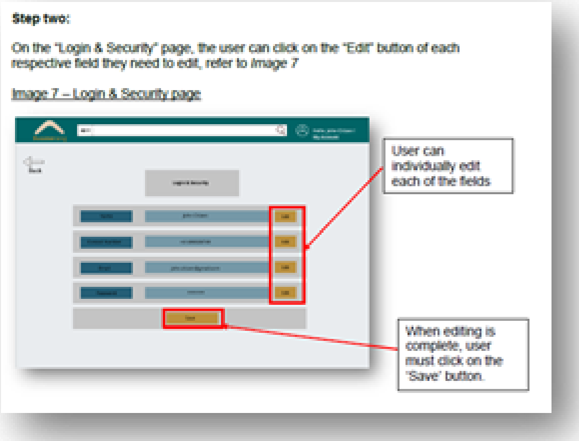
2.1.2. Login & Security

This is a sub page of the "User Settings" page. Users can use it to edit and/or update their login and security information, user attributes that can be edited in this interface are "Name", "Contact Number", "Email" and "Password".

When stepping through the process of updating information, users are prompted before confirming any changes to their data. Refer to process diagram in image 5.

Image 5 – Login & Security diagram

Image 2



Step two:

On the "Login & Security" page, the user can click on the "Edit" button of each respective field they need to edit, refer to *Image 7*

Image 7 – Login & Security page

User can individually edit each of the fields

When editing is complete, user must click on the 'Save' button.

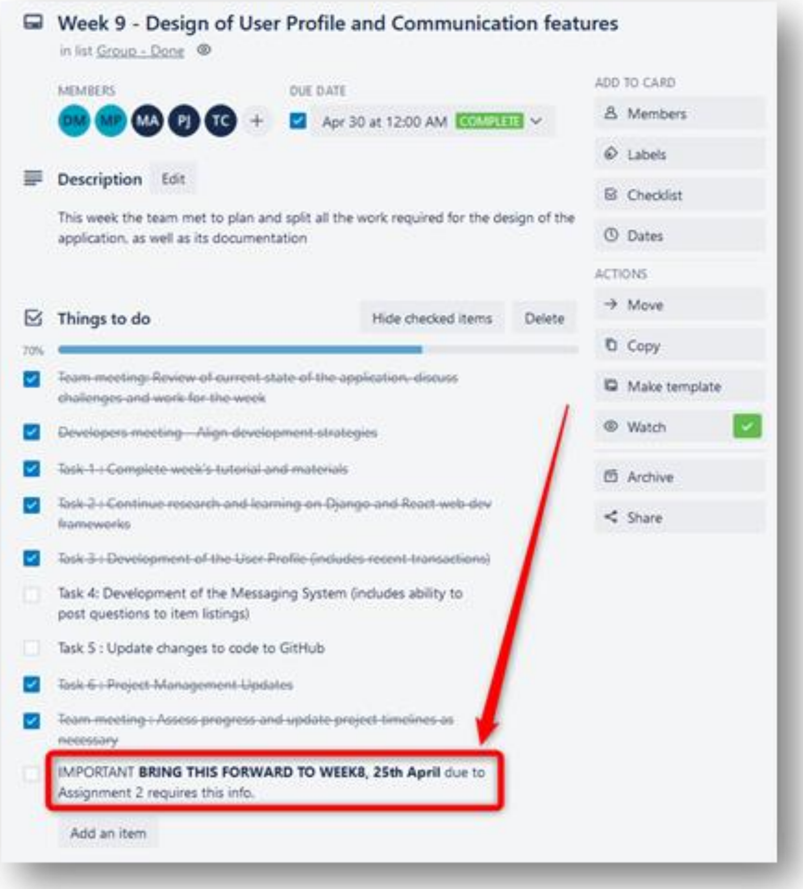
The weeks 6 to 8 were probably the 3 most intensive and labour intense weeks of the course to this date, during these weeks we realised and learned some especially important things, these were:

- How challenging it is to have "multiple cooks", in other words, to have multiple people designing different parts of the applications and expect that all pieces of the design would simply fit with each other.
- How to put together a framework and guidelines to address the previous stated challenge.
- How much documentation it takes to concisely explain how a seemingly simple feature works.
- How to get multiple project stake holders to agree on minor complexities and variations of the system and move forward.
- How the human aspect of a project as just as important as the technical ones.

As a note, during week 8 we also realised how some of our original dates estimates were not quite aligned with the dates we were supposed to submit the interim project report together with the system design. An example of this is the fact we were meant to submit the Assignment 2 by the end of week 8 (25th of April), while in our original Trello board we were still yet to complete the design of 2 of the minimum viable features ("Search Function" and "Add Items"). And yet a larger oversight was the fact that the design of the minimum viable features "User Profile" and "Communication" were only scheduled to be started the week after the submission of the Assignment 2.

Clearly, we would not have been able to submit Assignment 2 without all the features design being complete (or mostly complete). We had to improvise and re-shuffle priorities, re-allocate certain tasks last minute and "juggle commitments around". Refer to Image 3 below, an example in Trello of an instance where this happened.

Image 3



Week 9 - Design of User Profile and Communication features

in list Group - Done

MEMBERS: DM MP MA PJ TC

DUE DATE: ☑ Apr 30 at 12:00 AM COMPLETE

Description

This week the team met to plan and split all the work required for the design of the application, as well as its documentation

Things to do

70%

- ☑ ~~Team meeting: Review of current state of the application, discuss challenges and work for the week~~
- ☑ ~~Developers meeting – Align development strategies~~
- ☑ ~~Task 1 : Complete week's tutorial and materials~~
- ☑ ~~Task 2 : Continue research and learning on Django and React web dev frameworks~~
- ☑ ~~Task 3 : Development of the User Profile (includes recent transactions)~~
- □ Task 4: Development of the Messaging System (includes ability to post questions to item listings)
- □ Task 5 : Update changes to code to GitHub
- ☑ ~~Task 6 : Project Management Updates~~
- ☑ ~~Team meeting : Assess progress and update project timelines as necessary~~
- □ IMPORTANT **BRING THIS FORWARD TO WEEK8, 25th April** due to Assignment 2 requires this info.

By the end of week 8, each team member had contributed their section of the high-fidelity Figma prototype, including interactive steps as well as their respective detailed functional diagrams and step-by-step documentation of their part of system.

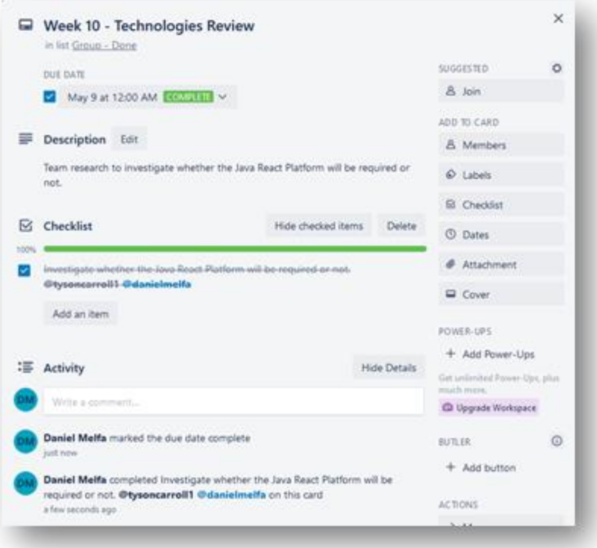
The team lead (Daniel Melfa) collated everyone's work in formatted everything to ensure consistent formatting. The assignment 2 was submitted on the 25th of April.

The assignment 2 was an important milestone for the group, from that point forward we would have more time to start "hand-on-work" in the technical aspects of the project, that is, spending time learning our chosen technologies and attempt to develop parts of the system.

The following weeks 9 and 10, the team delved into some investigation into whether the 2 web development platforms we initially chose "Django" and "React" would indeed be required for the tasks ahead. Our team members had none to minimal prior experience with programming languages and web development, having to learn two different frameworks from the ground up rapidly became a hurdle that we significantly underestimated.

While completing other parts of the course including the micro-credentials, learning how to use Kanban boards (Trello), Lucid charts, Prototyping tools (Figma), and completing the assignments were all considered by the group as something we should be able to achieve with reasonable efforts, the team somewhat underestimated the complexities of learning new development frameworks and doing something for the first time.

Some of the late changes we decided to do at week 10 was to drop the “Java React framework” from our list of chosen technologies. Upon extensive discussion and research, we agreed that we would be able to develop the interfaces of our website using plain HTML and the Django framework alone. The data would be hosted using the relational database MySQL. This was documented in Trello in a card tagged for Week10.



Week 10 - Technologies Review

in list Group - Done

DUE DATE

☑ May 9 at 12:00 AM COMPLETE

Description

Team research to investigate whether the Java React Platform will be required or not.

Checklist

100%

☑ ~~Investigate whether the Java React Platform will be required or not.~~ ~~@tysoncarroll1 @danielmelfa~~

Activity

Daniel Melfa marked the due date complete

just now

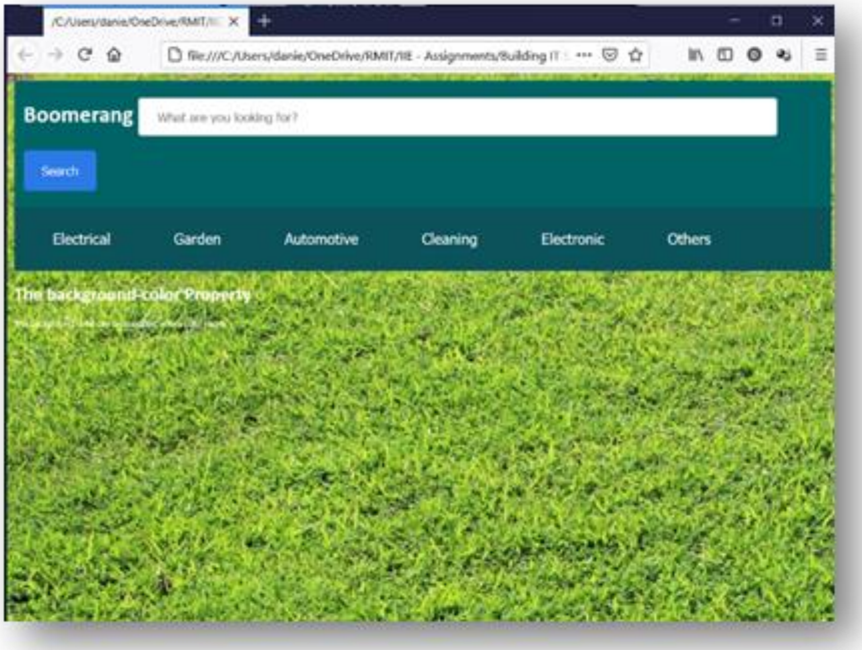
Daniel Melfa completed Investigate whether the Java React Platform will be required or not. @tysoncarroll1 @danielmelfa on this card

a few seconds ago

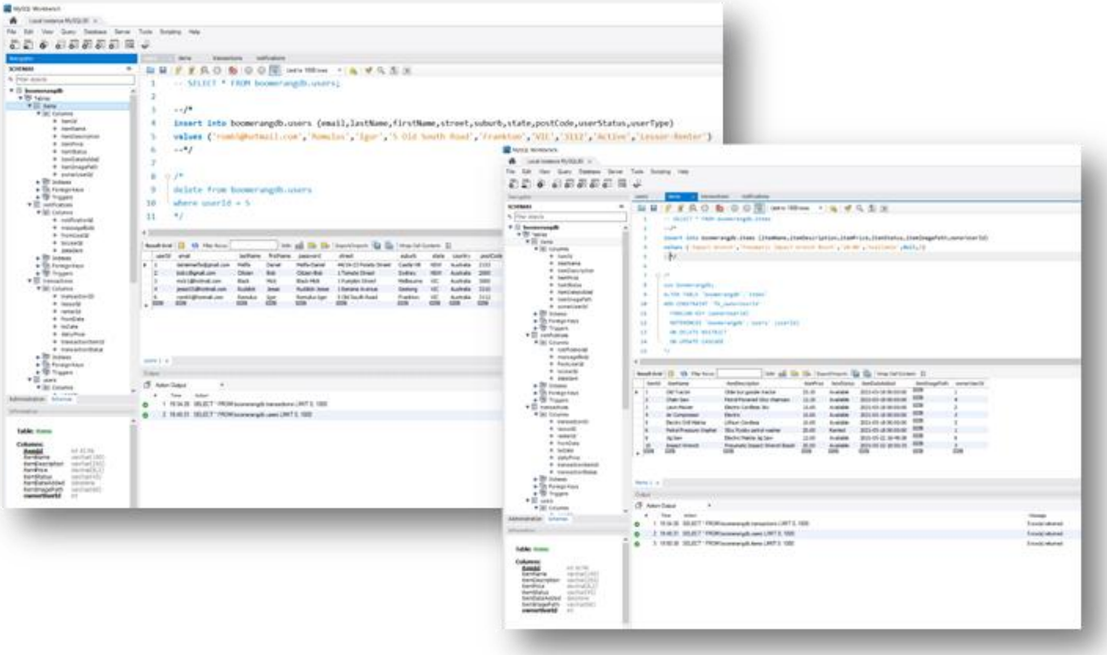
In the following weeks 10 and 11 the team developed some foundation components of the system, including some Django framework user authentication system and the relational database in MySQL to store all the information required by the system.

Some of the artifacts developed include:

The homepage, designed by Patrick Jenner


/C:/Users/danie/OneDrive/RMIT/II
file:///C:/Users/danie/OneDrive/RMIT/IIE - Assignments/Building IT
Boomerang
What are you looking for?
Search
Electrical
Garden
Automotive
Cleaning
Electronic
Others
The background-color Property

The relational database schema and tables, designed by Daniel Melfa



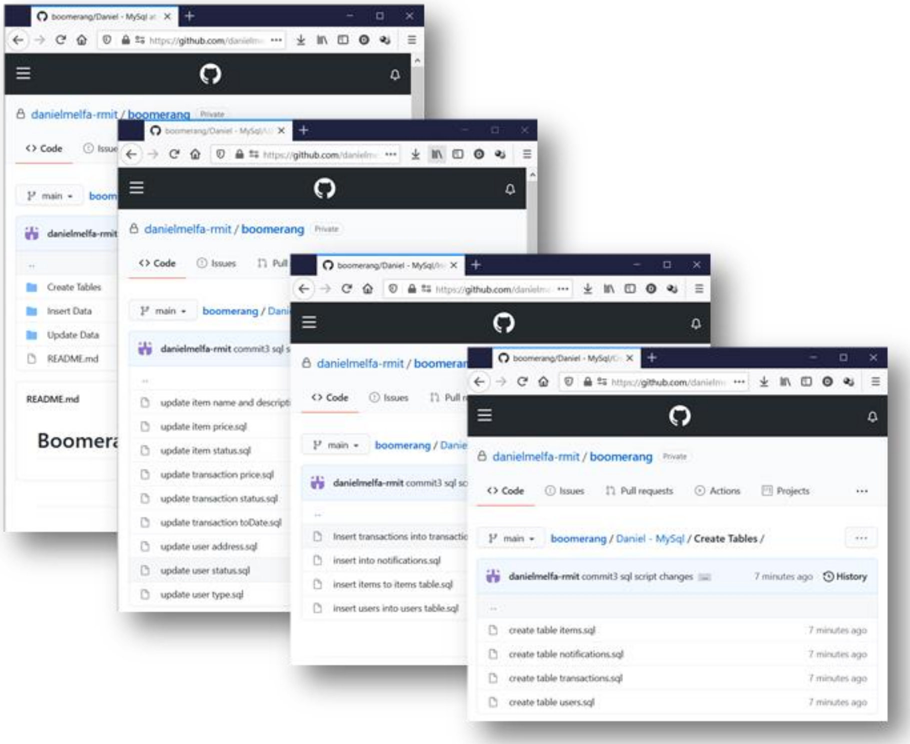
```
-- SELECT * FROM boomerangdb.users;

--/*
insert into boomerangdb.users (email,lastName,firstName,street,suburb,state,postCode,userStatus,userType)
values ('romk@hotmail.com','Romulus','Igor','5 Old South Road','Frankton','VIC','3112','Active','Lessor-Renter')
--*/

/*
delete from boomerangdb.users
where userId = 5
*/
```

The MySQL code repository in GitHub, designed by Daniel Melfa

The GitHub repository for the Boomerang web application can be found on this link: https://github.com/danielmelfa-rmit/boomerang



danielmelfa-rmit / boomerang Private

main boomerang / Daniel - MySql

danielmelfa-rmit

- Create Tables
- Insert Data
- Update Data
- README.md

README.md

Boomera

danielmelfa-rmit commit3 sql s

- update item name and descripti
- update item price.sql
- update item status.sql
- update transaction price.sql
- update transaction status.sql
- update transaction toDate.sql
- update user address.sql
- update user status.sql
- update user type.sql

danielmelfa-rmit commit3 sql sc

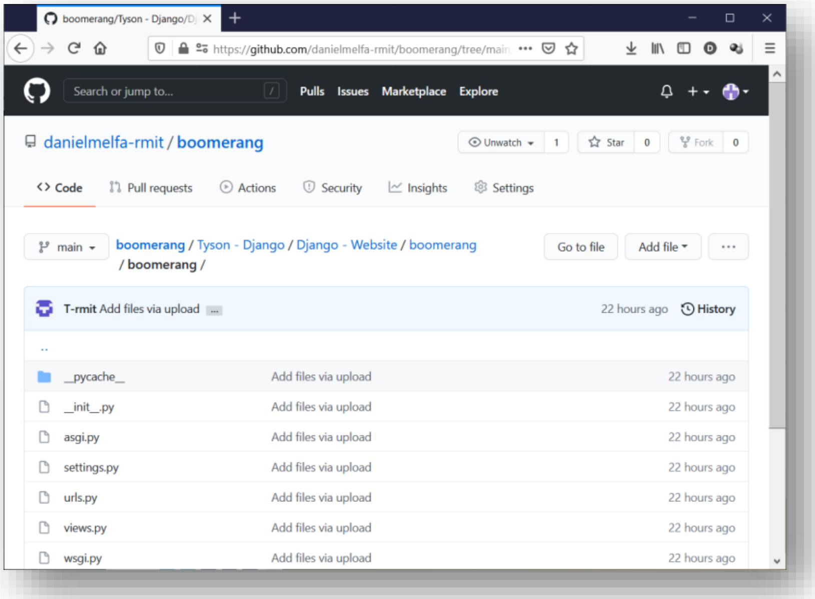
- Insert transactions into transactio
- insert into notifications.sql
- insert items to items table.sql
- insert users into users table.sql

main boomerang / Daniel - MySql / Create Tables /

danielmelfa-rmit commit3 sql script changes 7 minutes ago History

create table items.sql	7 minutes ago
create table notifications.sql	7 minutes ago
create table transactions.sql	7 minutes ago
create table users.sql	7 minutes ago

The Django code repository in GitHub, designed by Tyson Carroll



https://github.com/danielmelfa-rmit/boomerang/tree/main

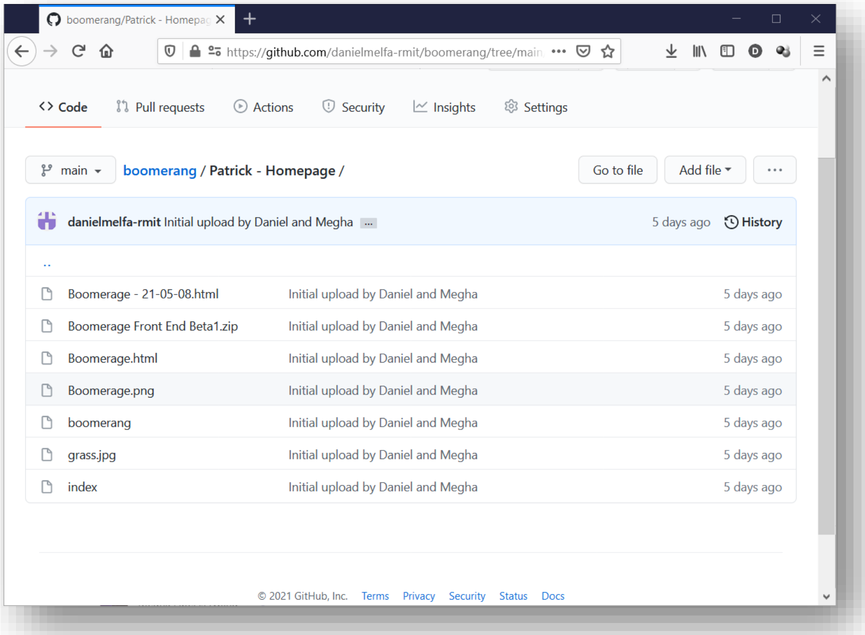
danielmelfa-rmit / boomerang

main boomerang / Tyson - Django / Django - Website / boomerang / boomerang /

T-rmit Add files via upload 22 hours ago History

__pycache__	Add files via upload	22 hours ago
__init__.py	Add files via upload	22 hours ago
asgi.py	Add files via upload	22 hours ago
settings.py	Add files via upload	22 hours ago
urls.py	Add files via upload	22 hours ago
views.py	Add files via upload	22 hours ago
wsgi.py	Add files via upload	22 hours ago

The Homepage HTML template code repository in GitHub, designed by Patrick Jenner

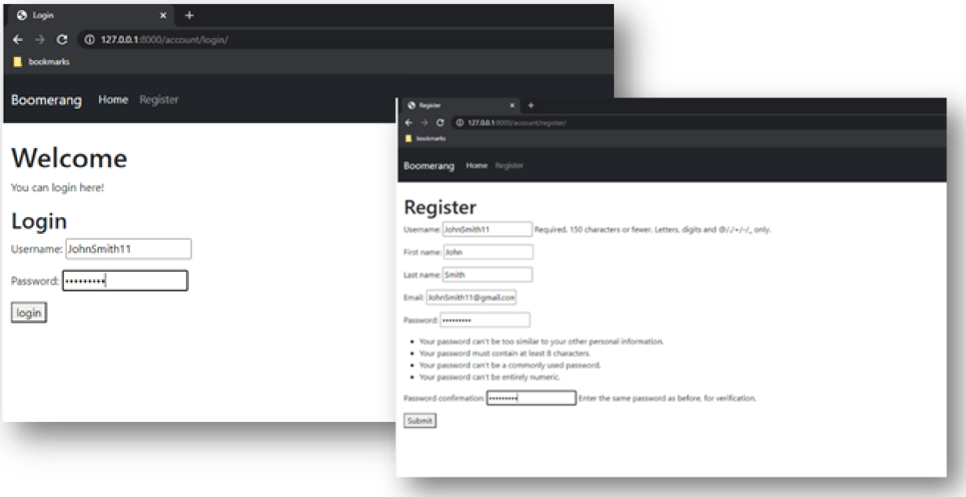


boomerang / Patrick - Homepage /

danielmelfa-rmit Initial upload by Daniel and Megha … 5 days ago

..		
Boomerage - 21-05-08.html	Initial upload by Daniel and Megha	5 days ago
Boomerage Front End Beta1.zip	Initial upload by Daniel and Megha	5 days ago
Boomerage.html	Initial upload by Daniel and Megha	5 days ago
Boomerage.png	Initial upload by Daniel and Megha	5 days ago
boomerang	Initial upload by Daniel and Megha	5 days ago
grass.jpg	Initial upload by Daniel and Megha	5 days ago
index	Initial upload by Daniel and Megha	5 days ago

The Django framework foundation work, including user authentication, designed by Tyson Carroll



Welcome

You can login here!

Login

Username: JohnSmith11

Password: ••••••••••

Register

Username: JohnSmith11 Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name: John

Last name: Smith

Email: JohnSmith11@gmail.con

Password: ••••••••••

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation: •••••••••• Enter the same password as before, for verification.

The Django framework internal SQL Lite database, designed by Tyson Carroll

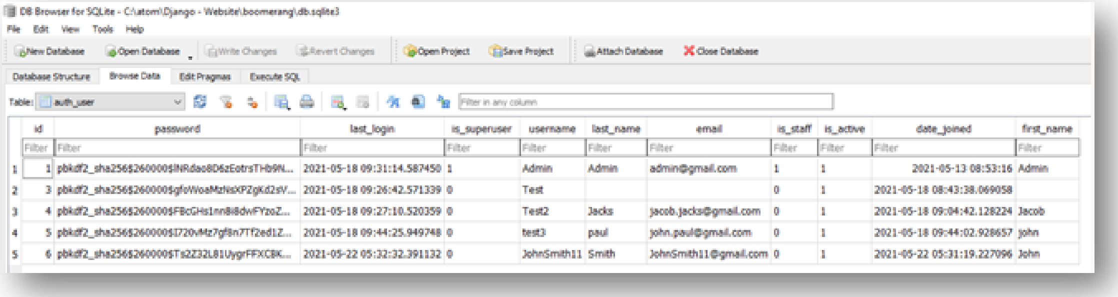
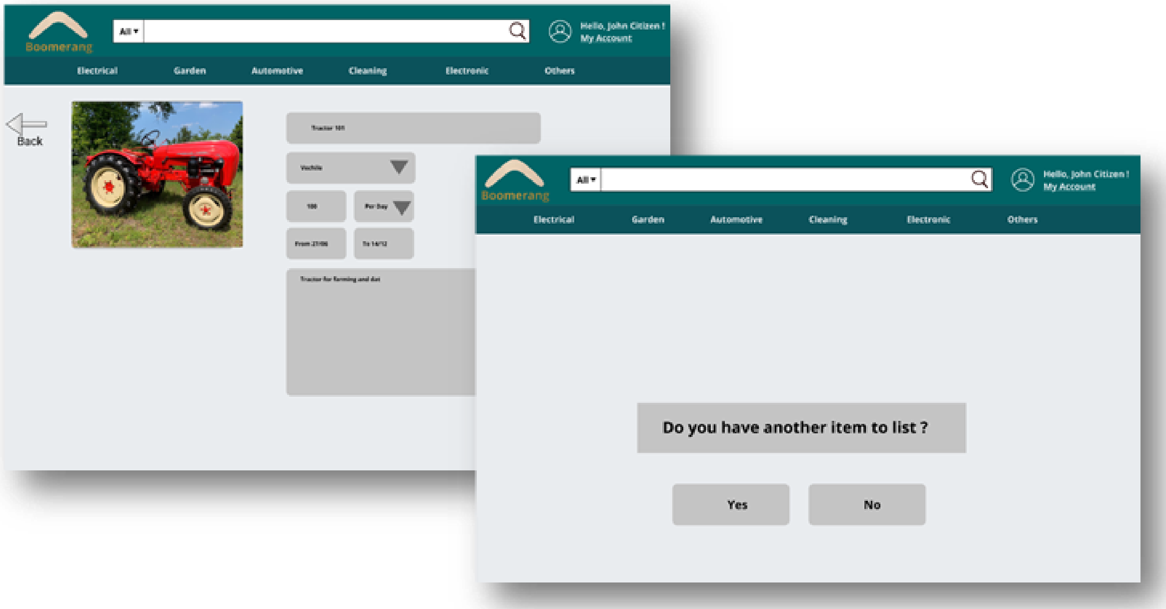


Table: auth_user

Filter in any column

	id	password	last_login	is_superuser	username	last_name	email	is_staff	is_active	date_joined	first_name
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	pbkdf2_sha256$260000$INRdao8D6zEotrsTHb9N...	2021-05-18 09:31:14.587450	1	Admin	Admin	admin@gmail.com	1	1	2021-05-13 08:53:16	Admin
2	3	pbkdf2_sha256$260000$gfoWoaMzNsXPZgKd2sV...	2021-05-18 09:26:42.571339	0	Test			0	1	2021-05-18 08:43:38.069058	
3	4	pbkdf2_sha256$260000$FBcGHs1nn8i8dwFYzoZ...	2021-05-18 09:27:10.520359	0	Test2	Jacks	jacob.jacks@gmail.com	0	1	2021-05-18 09:04:42.128224	Jacob
4	5	pbkdf2_sha256$260000$I720vMz7gf8n7Tf2ed1Z...	2021-05-18 09:44:25.949748	0	test3	paul	john.paul@gmail.com	0	1	2021-05-18 09:44:02.928657	john
5	6	pbkdf2_sha256$260000$Ts2Z32L81UygrFFXCBK...	2021-05-22 05:32:32.391132	0	JohnSmith11	Smith	JohnSmith11@gmail.com	0	1	2021-05-22 05:31:19.227096	John

Prototype “Add new item for lease”, designed by Moustafa Al-Meahi



Tractor 101

Vechile

100

Per Day

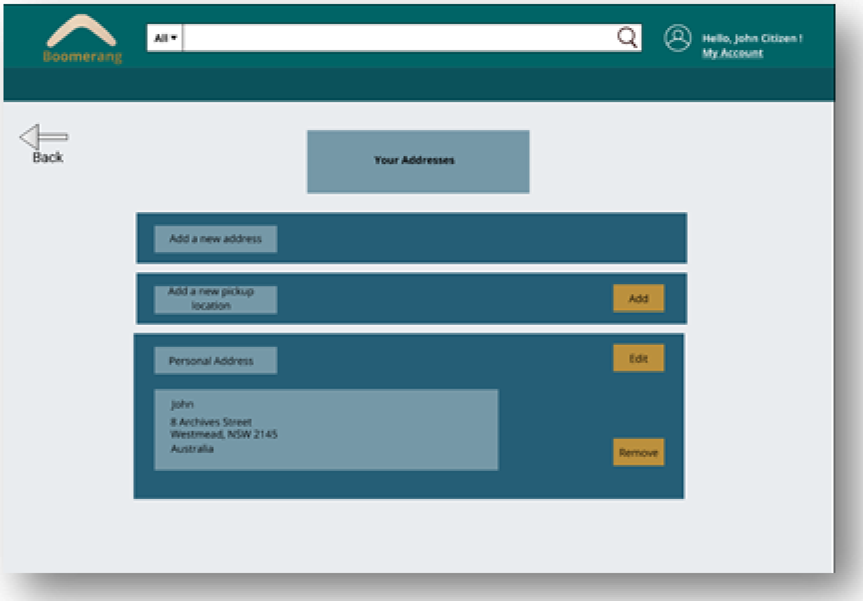
From 21/06

To 14/12

Tractor for farming and dirt

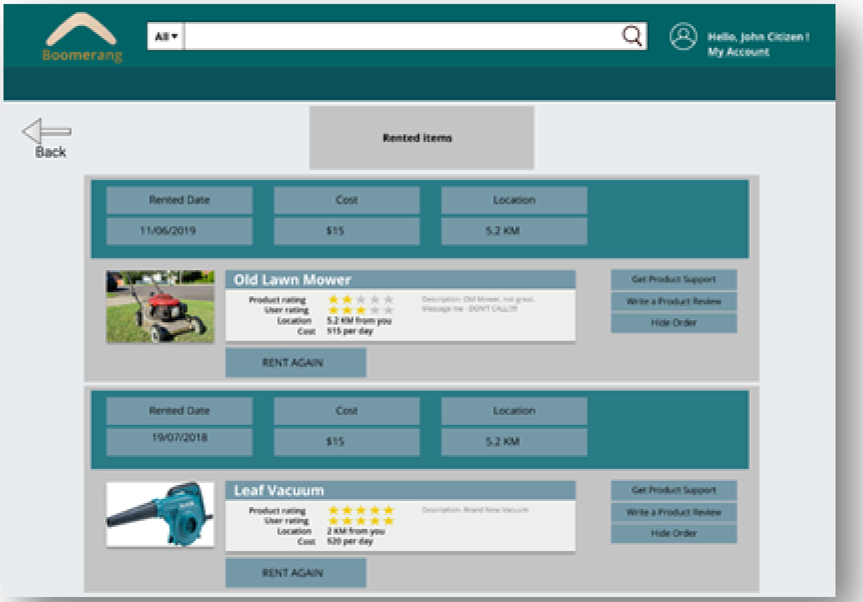
Do you have another item to list ?

Prototype “User settings”, designed by Megha Patel



John
8 Archives Street
Westmead, NSW 2145
Australia

Prototype “Rented Items”, designed by Megha Patel



Rented Date	Cost	Location
11/06/2019	$15	5.2 KM

Old Lawn Mower

Product rating	★★☆☆☆
User rating	★★★☆☆
Location	5.2 KM from you
Cost	$15 per day

Description: Old Mower, not great. Message me - DONT CALL!!!

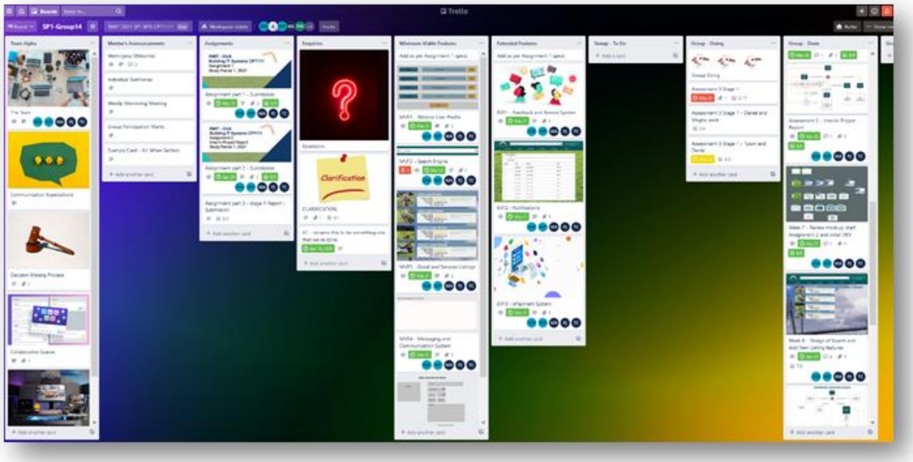
Rented Date	Cost	Location
19/07/2018	$15	5.2 KM

Leaf Vacuum

Product rating	★★★★★
User rating	★★★★★
Location	2 KM from you
Cost	$20 per day

Description: Brand New Vacuum

Group 14 Trello board, managed by all team members



Link https://trello.com/b/SG3eax4b/sp1-group14
As of the date of authoring this document, the Group 14 got as far as having the system architected, documented, including a high-fidelity **interactive** FIGMA prototype (refer to link below) and some of the core components of the system including the foundation configuration of user authentication in the Django framework, a HTML initial homepage and a MySQL database with the data.

FIGMA LINK:
https://www.figma.com/proto/vJbJ5IcnhHjUTmPLraDQhx/Boomerang?node-id=1%3A2&scaling=min-zoom&page-id=0%3A1

These components are not integrated with each other, research is still ongoing on how to get these components functioning together, however, given the timeframes ahead the chances of having a functional system fully integrated is considered to be outside of the group current delivery capabilities.

Some key aspects of this project that the team sees as an oversight and would like to have spent more time on would be to have researched further on the integration aspects of the key technologies chosen, in other words, how to integrate web pages written in HTML in the Django framework, how to embed Python portions of code into a HTML page, and how to make Django talk to a MySQL database instead of its internal SQL Lite database that comes embed with it.

In sum, if we had been able to figure-out these integration technical challenges earlier in the project, we could possibly have completed some integration of these disparate pieces of work the team developed.

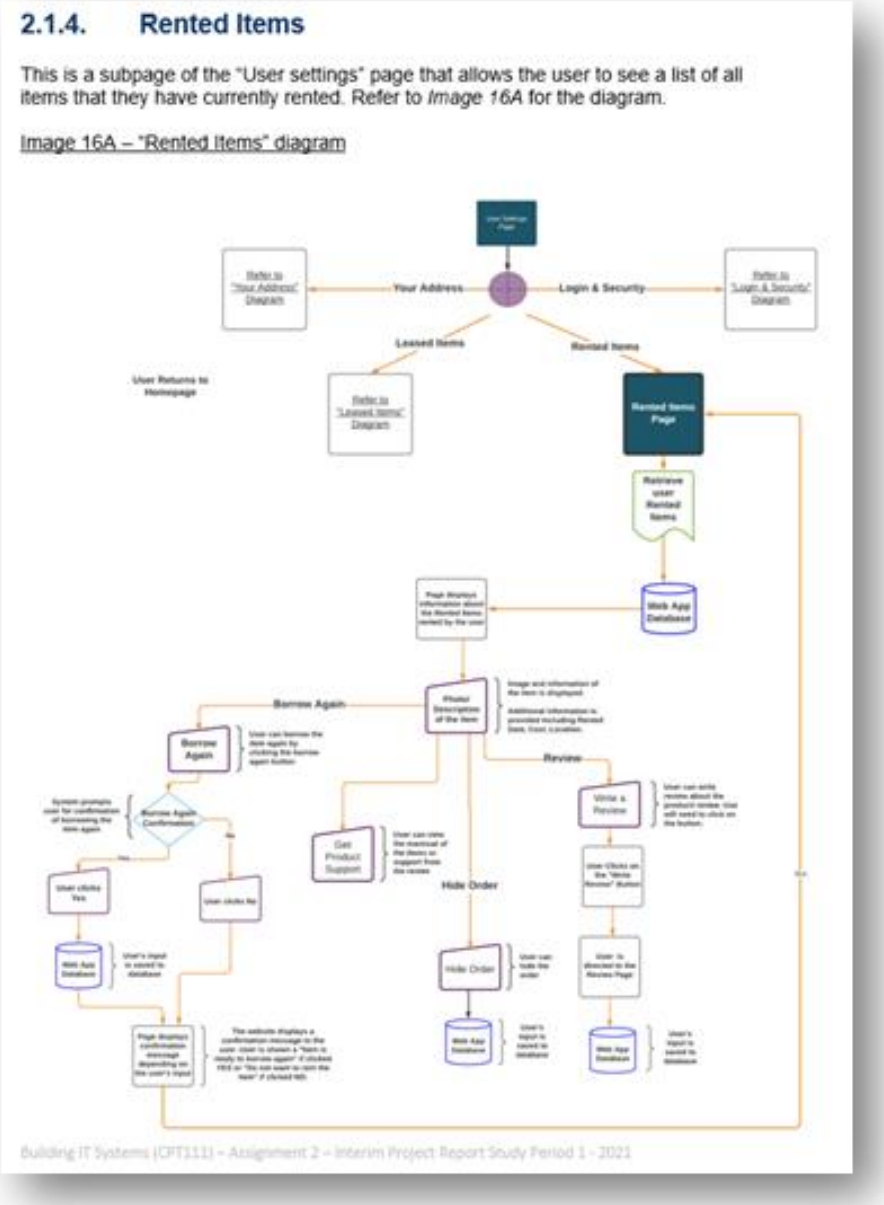
2.2. Outcomes to date

System Architecture and Design

The groundwork including detailed system documentation, in-depth functional diagrams, step-by-step guides, and validation use cases have been thoroughly designed to make all features of the system unambiguous and to allow a web developer to implement it without much guess work. The complete solution design was submitted as part of the Assignment 2, the Interim Project Report.

Our minimum viable features from 1 to 5 were all grade 9.6 points out of a possible 12, that indicates the quality and level of details that were included.

Image 1 - Example of in-depth functional diagrams for feature “Rented Items”:



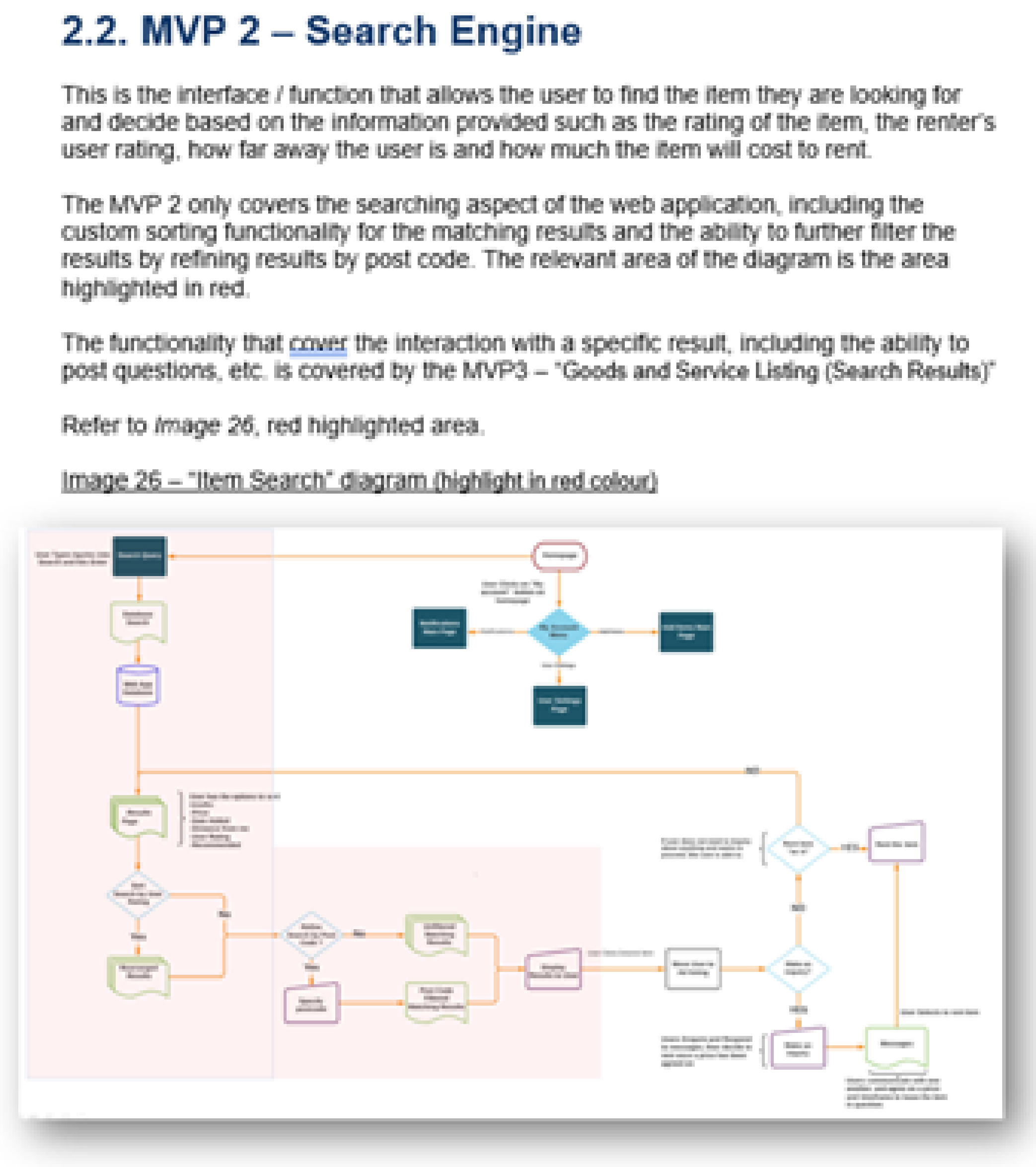
2.1.4. Rented Items

This is a subpage of the “User settings” page that allows the user to see a list of all items that they have currently rented. Refer to *Image 16A* for the diagram.

Image 16A – “Rented Items” diagram

Building IT Systems (CPT111) – Assignment 2 – Interim Project Report Study Period 1 - 2021

Image 2 – Example of in-depth functional diagrams for feature “Rented Items”:



2.2. MVP 2 – Search Engine

This is the interface / function that allows the user to find the item they are looking for and decide based on the information provided such as the rating of the item, the renter's user rating, how far away the user is and how much the item will cost to rent.

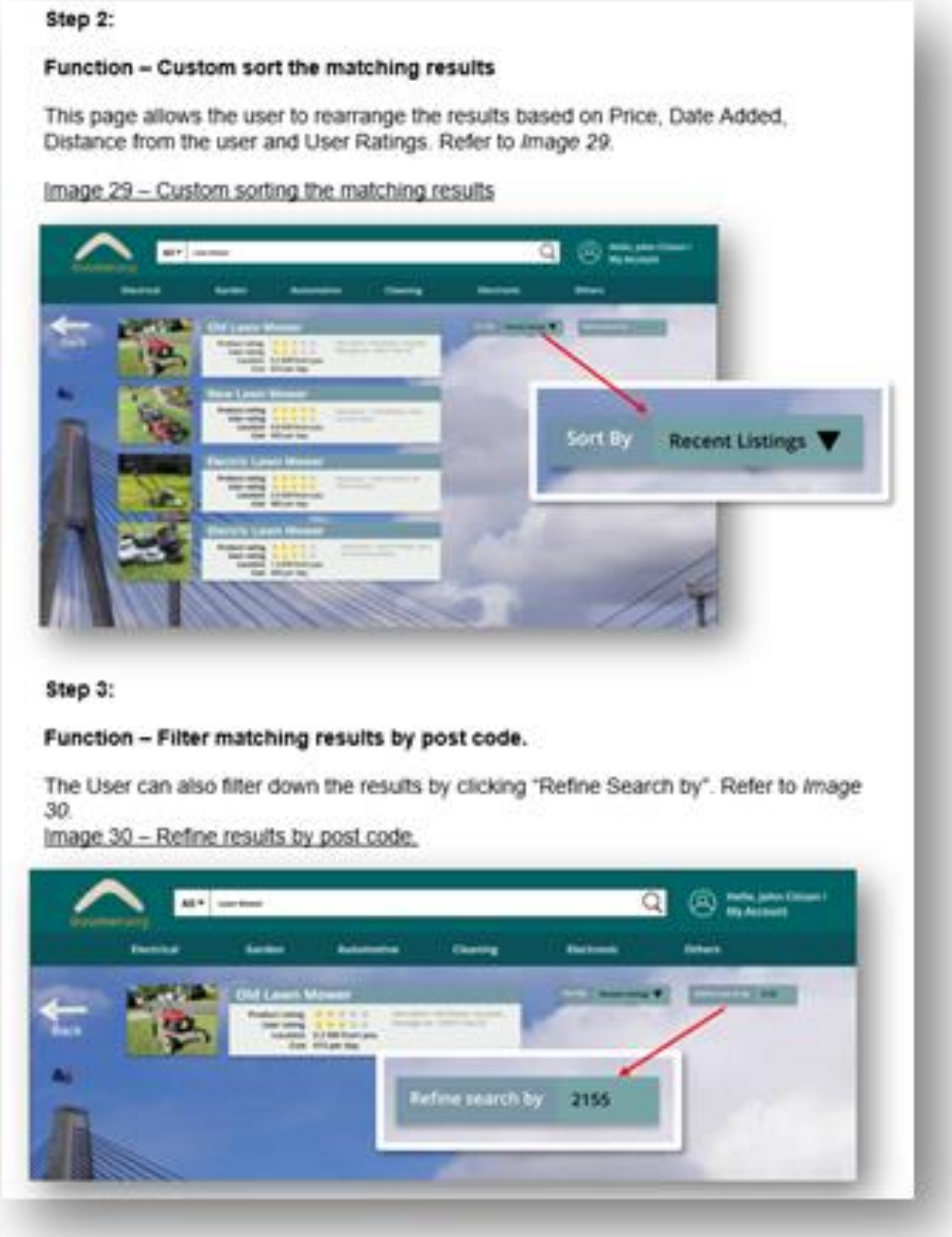
The MVP 2 only covers the searching aspect of the web application, including the custom sorting functionality for the matching results and the ability to further filter the results by refining results by post code. The relevant area of the diagram is the area highlighted in red.

The functionality that cover the interaction with a specific result, including the ability to post questions, etc. is covered by the MVP3 – "Goods and Service Listing (Search Results)"

Refer to *Image 26*, red highlighted area.

Image 26 – "Item Search" diagram (highlight in red colour)

Image 3 - Example of comprehensive step-by-step guides of system functionality



Step 2:

Function – Custom sort the matching results

This page allows the user to rearrange the results based on Price, Date Added, Distance from the user and User Ratings. Refer to *Image 29*.

Image 29 – Custom sorting the matching results

Step 3:

Function – Filter matching results by post code.

The User can also filter down the results by clicking "Refine Search by". Refer to *Image 30*.

Image 30 – Refine results by post code.

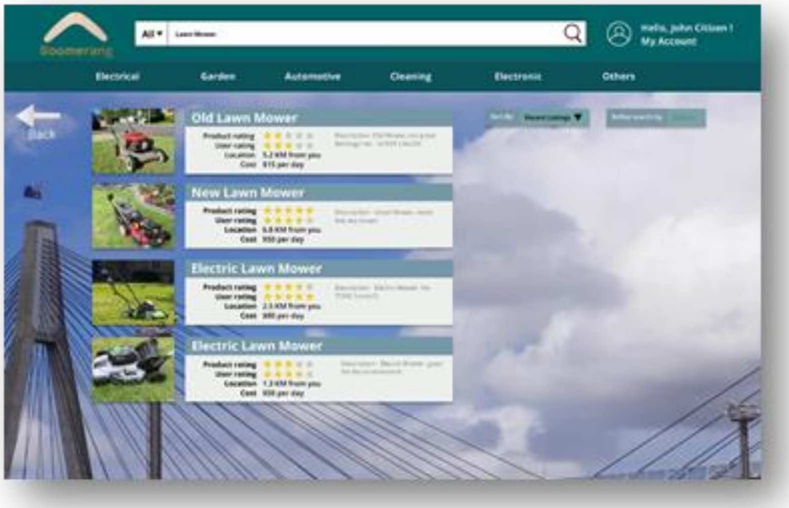
High-fidelity Interactive FIGMA prototype

https://www.figma.com/proto/vJbJ5IcnhHjUTmPLraDQhx/Boomerang?node-id=1%3A2&scaling=min-zoom&page-id=0%3A1

The team decided that creating a high-fidelity Interactive prototype **including all the proposed features** of the application would be notably beneficial. Not only having an interactive prototype makes it easier for us to "sell
the idea" to future investors, and it also facilitates the demonstration of the proposed solution to other peers.

In additional, having a high-fidelity prototype serves as guidance for the technical developers to design the solution exactly as envisaged by its designers and architects.

Image 1 - Interactive FIGMA prototype, "Search Results":



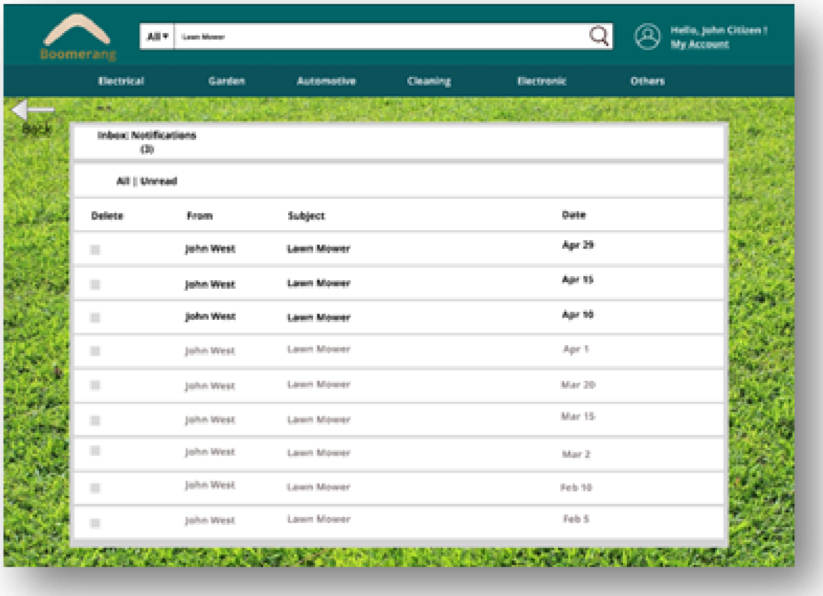
Old Lawn Mower
Product rating ★★☆☆☆
User rating ★★★☆☆
Location 5.2 KM from you
Cost $15 per day

New Lawn Mower
Product rating ★★★★★
User rating ★★★★☆
Location 6.8 KM from you
Cost $50 per day

Electric Lawn Mower
Product rating ★★★★☆
User rating ★★★★★
Location 2.3 KM from you
Cost $80 per day

Electric Lawn Mower
Product rating ★★★☆☆
User rating ★★★★☆
Location 1.3 KM from you
Cost $50 per day

Image 2 - Interactive FIGMA prototype, "Notification System":

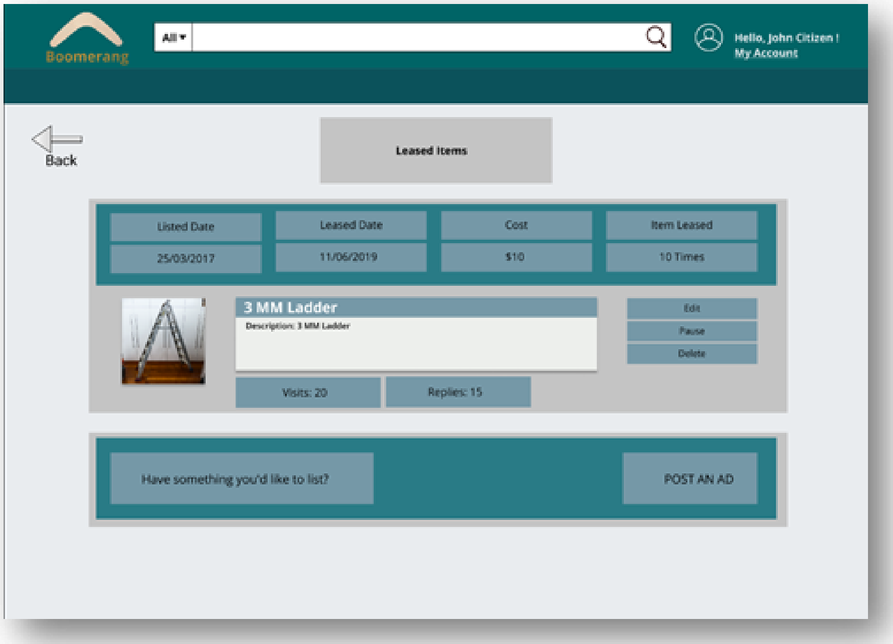


Inbox: Notifications
(3)

All | Unread

Delete	From	Subject	Date
□	John West	Lawn Mower	Apr 29
□	John West	Lawn Mower	Apr 15
□	John West	Lawn Mower	Apr 10
□	John West	Lawn Mower	Apr 1
□	John West	Lawn Mower	Mar 20
□	John West	Lawn Mower	Mar 15
□	John West	Lawn Mower	Mar 2
□	John West	Lawn Mower	Feb 10
□	John West	Lawn Mower	Feb 5

Image 3 - Interactive FIGMA prototype, “My Leased Items”:



Listed Date	Leased Date	Cost	Item Leased
25/03/2017	11/06/2019	$10	10 Times

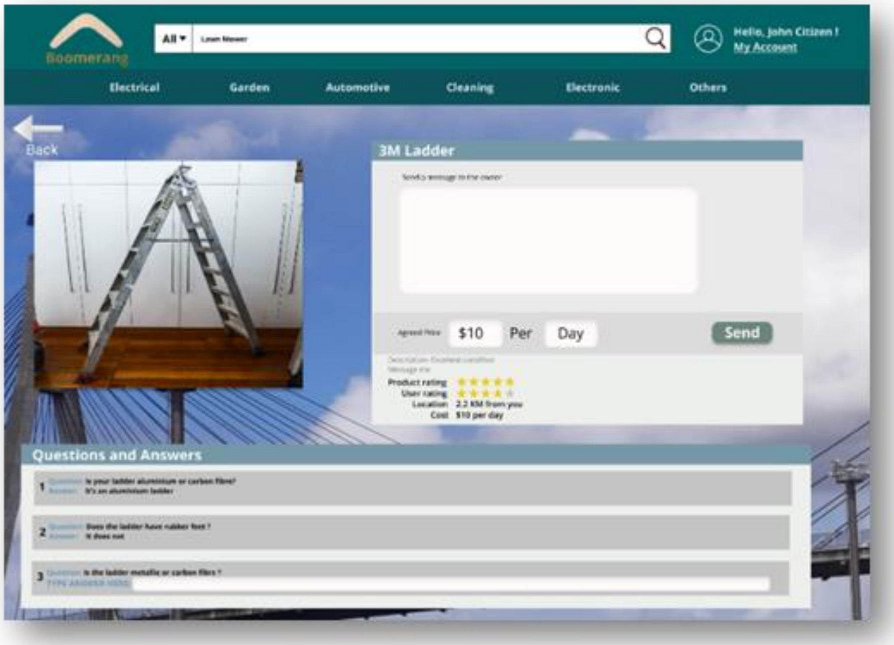
3 MM Ladder

Description: 3 MM Ladder

Visits: 20

Replies: 15

Image 4 - Interactive FIGMA prototype, “My Rented Items”, including feature to respond to user inquiries:



3M Ladder

Send a message to the owner

Agreed Price $10 Per Day

Description: Excellent condition

Message me

Product rating ★★★★★

User rating ★★★★☆

Location: 2.2 KM from you

Cost $10 per day

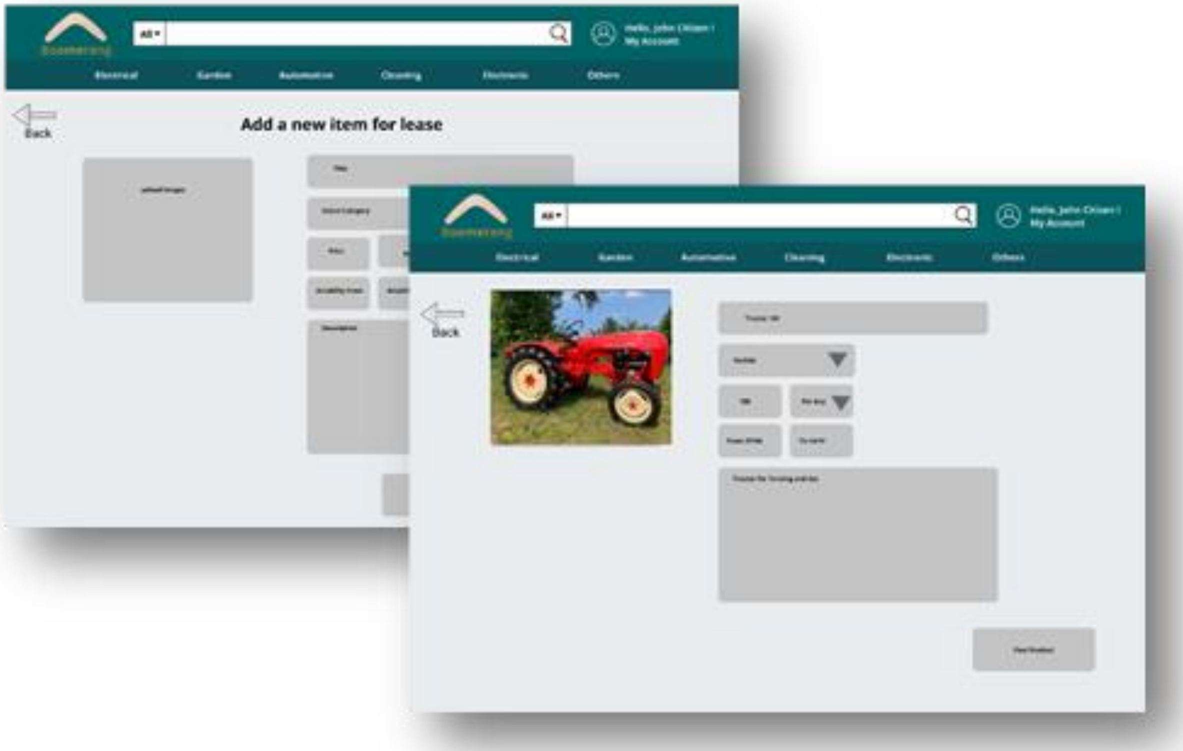
Questions and Answers

1 Question: Is your ladder aluminium or carbon fibre?
Answer: It's an aluminium ladder

2 Question: Does the ladder have rubber feet ?
Answer: It does not

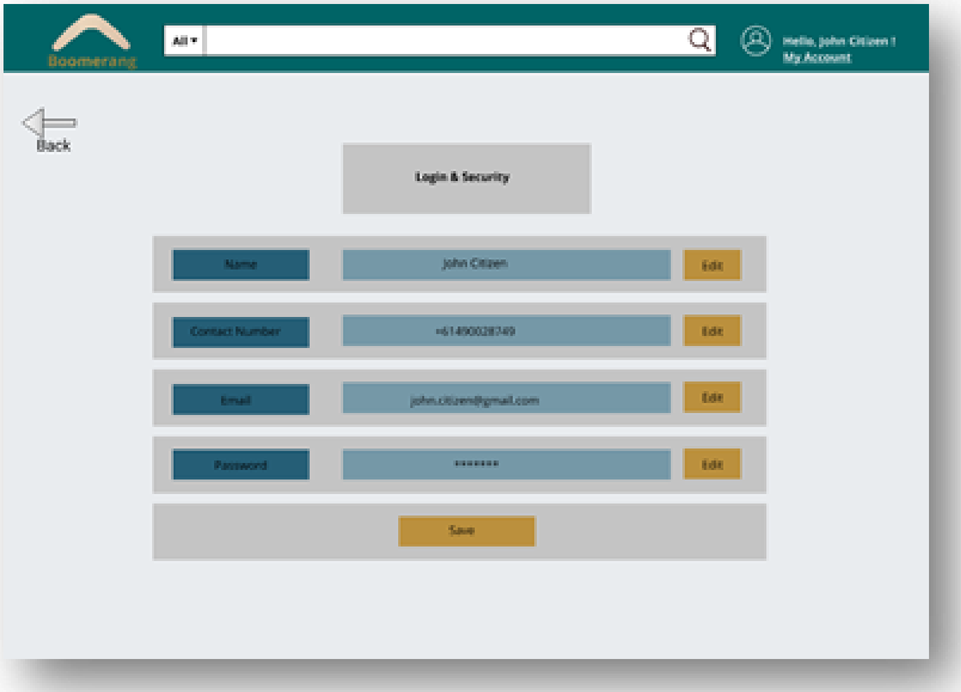
3 Question: Is the ladder metallic or carbon fibre ?
TYPE ANSWER HERE

Image 5 - Interactive FIGMA prototype, “Add new item for lease”:



Add a new item for lease

Image 6 - Interactive FIGMA prototype, “Login & Security” settings:

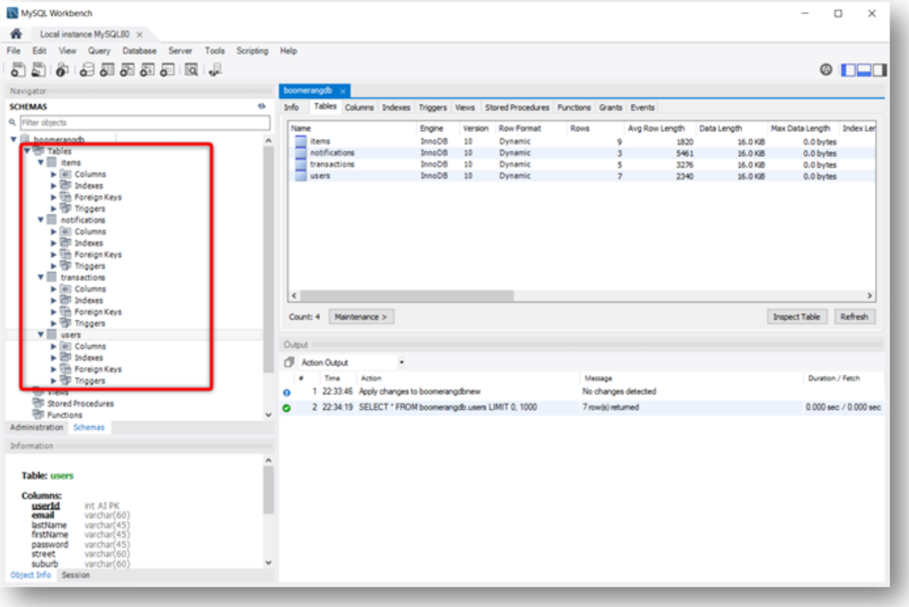


Login & Security

Name	John Citizen	Edit
Contact Number	+61490028749	Edit
Email	john.citizen@gmail.com	Edit
Password	*******	Edit

Relational database for the Web application, developed using MySQL

The relational data that stores the transaction details, items listed on the system, notifications and user information is stored in a MySQL database.



Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length	Max Data Length
items	InnoDB	10	Dynamic	9	1820	16.0 KiB	0.0 bytes
notifications	InnoDB	10	Dynamic	3	5461	16.0 KiB	0.0 bytes
transactions	InnoDB	10	Dynamic	5	3276	16.0 KiB	0.0 bytes
users	InnoDB	10	Dynamic	7	2340	16.0 KiB	0.0 bytes

Count: 4

Output — Action Output

```
#  Time      Action                                        Message               Duration / Fetch
1  22:33:46  Apply changes to boomerangdbnew               No changes detected
2  22:34:19  SELECT * FROM boomerangdb.users LIMIT 0, 1000 7 row(s) returned     0.000 sec / 0.000 sec
```

Information

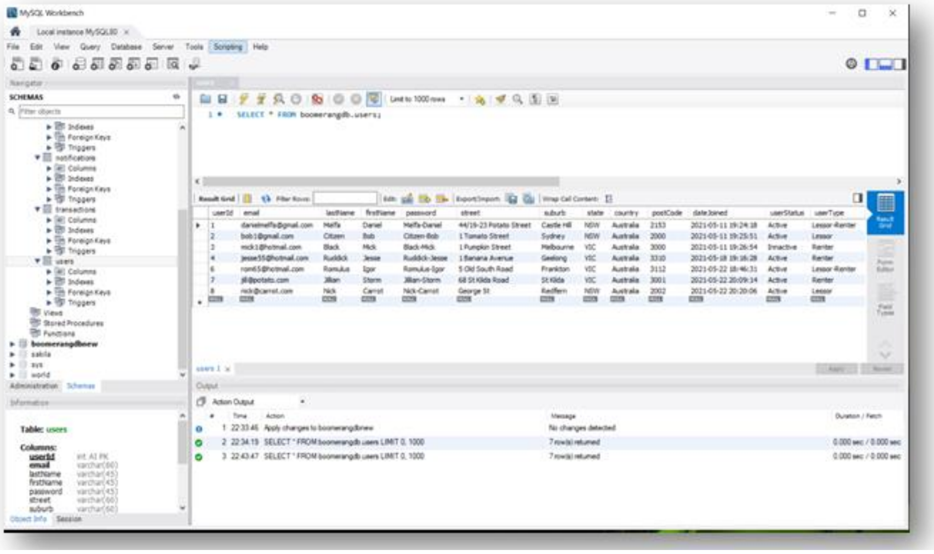
Table: users

Columns:
```
userId     int AI PK
email      varchar(60)
lastName   varchar(45)
firstName  varchar(45)
password   varchar(45)
street     varchar(60)
suburb     varchar(60)
```

Note: User information related to the Django framework authentication mechanism is currently stored within the Django own's embedded SQL Lite database.

Currently the schema for the "BoomerangDB" has 4 relational tables, these are all connected to each other by means of Foreign Key constrains. All table have been designed using the correct data types, all tables have their Primary keys defined. Foreign keys and constrains are extensively used throughout all tables to ensure data consistency.

The design of each table is as follows:



```
1 • SELECT * FROM boomerangdb.users;
```

userId	email	lastName	firstName	password	street	suburb	state	country	postCode	dateJoined	userStatus	userType
1	danielmelfa@gmail.com	Melfa	Daniel	Melfa-Daniel	44/19-23 Potato Street	Castle Hill	NSW	Australia	2153	2021-05-11 19:24:18	Active	Lessor-Renter
2	bob1@gmail.com	Citizen	Bob	Citizen-Bob	1 Tomato Street	Sydney	NSW	Australia	2000	2021-05-11 19:25:51	Active	Lessor
3	mick1@hotmail.com	Black	Mick	Black-Mick	1 Pumpkin Street	Melbourne	VIC	Australia	3000	2021-05-11 19:26:54	Inactive	Renter
4	jesse55@hotmail.com	Ruddick	Jesse	Ruddick-Jesse	1 Banana Avenue	Geelong	VIC	Australia	3310	2021-05-18 19:16:28	Active	Renter
6	rom65@hotmail.com	Romulus	Igor	Romulus-Igor	5 Old South Road	Frankton	VIC	Australia	3112	2021-05-22 18:46:31	Active	Lessor-Renter
7	jill@potato.com	Jillian	Storm	Jillian-Storm	68 St Kilda Road	St Kilda	VIC	Australia	3001	2021-05-22 20:09:14	Active	Renter
8	nick@carrot.com	Nick	Carrot	Nick-Carrot	George St	Redfern	NSW	Australia	2002	2021-05-22 20:20:06	Active	Lessor
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Output — Action Output

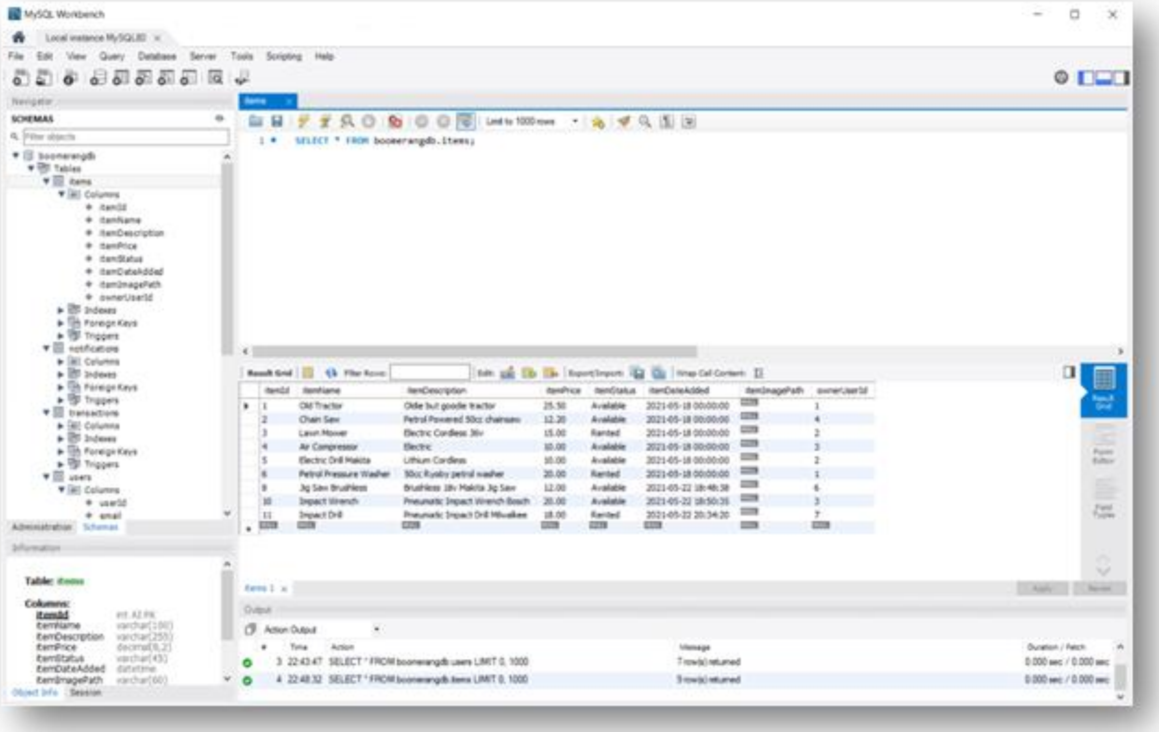
```
#  Time      Action                                        Message               Duration / Fetch
1  22:33:46  Apply changes to boomerangdbnew               No changes detected
2  22:34:19  SELECT * FROM boomerangdb.users LIMIT 0, 1000 7 row(s) returned     0.000 sec / 0.000 sec
3  22:43:47  SELECT * FROM boomerangdb.users LIMIT 0, 1000 7 row(s) returned     0.000 sec / 0.000 sec
```

Information

Table: users

Columns:
```
userId     int AI PK
email      varchar(60)
lastName   varchar(45)
firstName  varchar(45)
password   varchar(45)
street     varchar(60)
suburb     varchar(60)
```

Users Table Data Structure



```
1 •  SELECT * FROM boomerangdb.items;
```

itemId	itemName	itemDescription	itemPrice	itemStatus	itemDateAdded	itemImagePath	ownerUserId
1	Old Tractor	Olde but goodie tractor	25.50	Available	2021-05-18 00:00:00	NULL	1
2	Chain Saw	Petrol Powered 50cc chainsaw	12.20	Available	2021-05-18 00:00:00	NULL	4
3	Lawn Mower	Electric Cordless 36v	15.00	Rented	2021-05-18 00:00:00	NULL	2
4	Air Compressor	Electric	10.00	Available	2021-05-18 00:00:00	NULL	3
5	Electric Drill Makita	Lithium Cordless	10.00	Available	2021-05-18 00:00:00	NULL	2
6	Petrol Pressure Washer	50cc Ryoby petrol washer	20.00	Rented	2021-05-18 00:00:00	NULL	1
9	Jig Saw Brushless	Brushless 18v Makita Jig Saw	12.00	Available	2021-05-22 18:48:38	NULL	6
10	Impact Wrench	Pneumatic Impact Wrench Bosch	20.00	Available	2021-05-22 18:50:35	NULL	3
11	Impact Drill	Pneumatic Impact Drill Milwalkee	18.00	Rented	2021-05-22 20:34:20	NULL	7
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table: items

Columns:

itemId	int AI PK
itemName	varchar(100)
itemDescription	varchar(255)
itemPrice	decimal(8,2)
itemStatus	varchar(45)
itemDateAdded	datetime
itemImagePath	varchar(60)

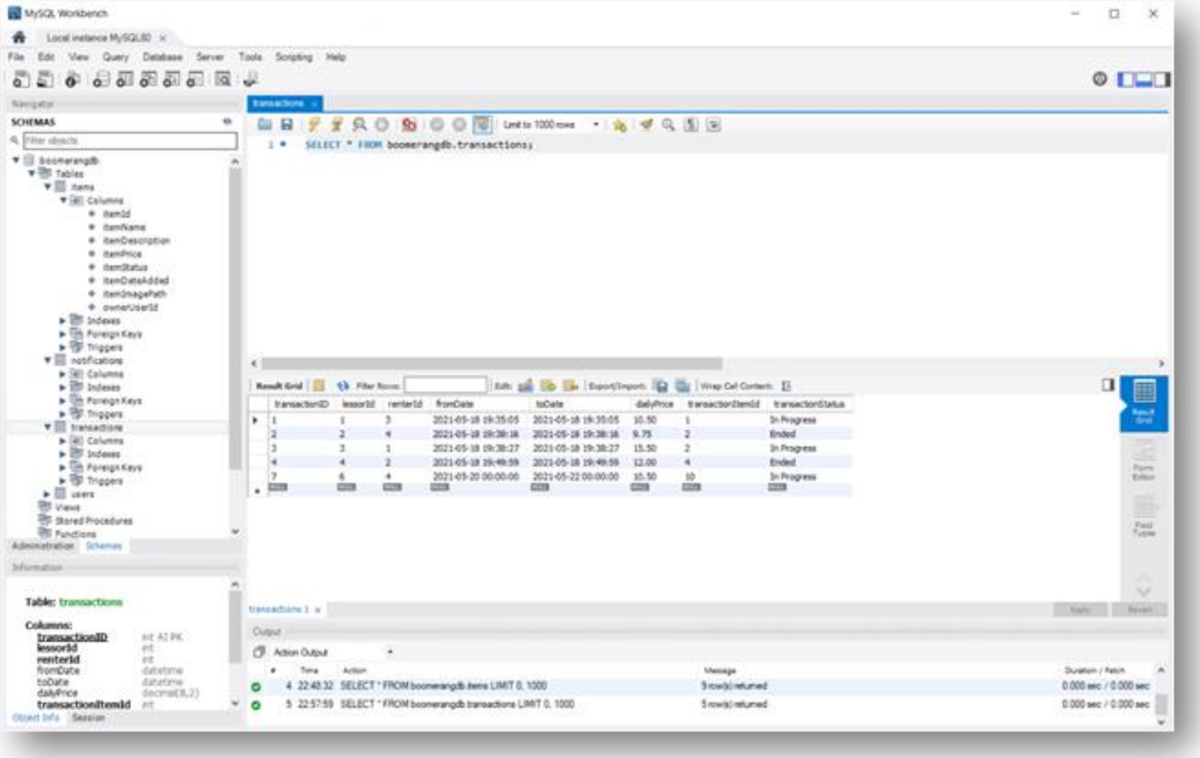
items 1

Output

Action Output

```
  #  Time      Action                                      Message              Duration / Fetch
  3  22:43:47  SELECT * FROM boomerangdb.users LIMIT 0, 1000  7 row(s) returned  0.000 sec / 0.000 sec
  4  22:48:32  SELECT * FROM boomerangdb.items LIMIT 0, 1000  9 row(s) returned  0.000 sec / 0.000 sec
```

Items Table Data Structure



```
1 •  SELECT * FROM boomerangdb.transactions;
```

transactionID	lessorId	renterId	fromDate	toDate	dailyPrice	transactionItemId	transactionStatus
1	1	3	2021-05-18 19:35:05	2021-05-18 19:35:05	10.50	1	In Progress
2	2	4	2021-05-18 19:38:16	2021-05-18 19:38:16	9.75	2	Ended
3	3	1	2021-05-18 19:38:27	2021-05-18 19:38:27	15.50	2	In Progress
4	4	2	2021-05-18 19:49:59	2021-05-18 19:49:59	12.00	4	Ended
7	6	4	2021-05-20 00:00:00	2021-05-22 00:00:00	10.50	10	In Progress
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table: transactions

Columns:

transactionID	int AI PK
lessorId	int
renterId	int
fromDate	datetime
toDate	datetime
dailyPrice	decimal(8,2)
transactionItemId	int

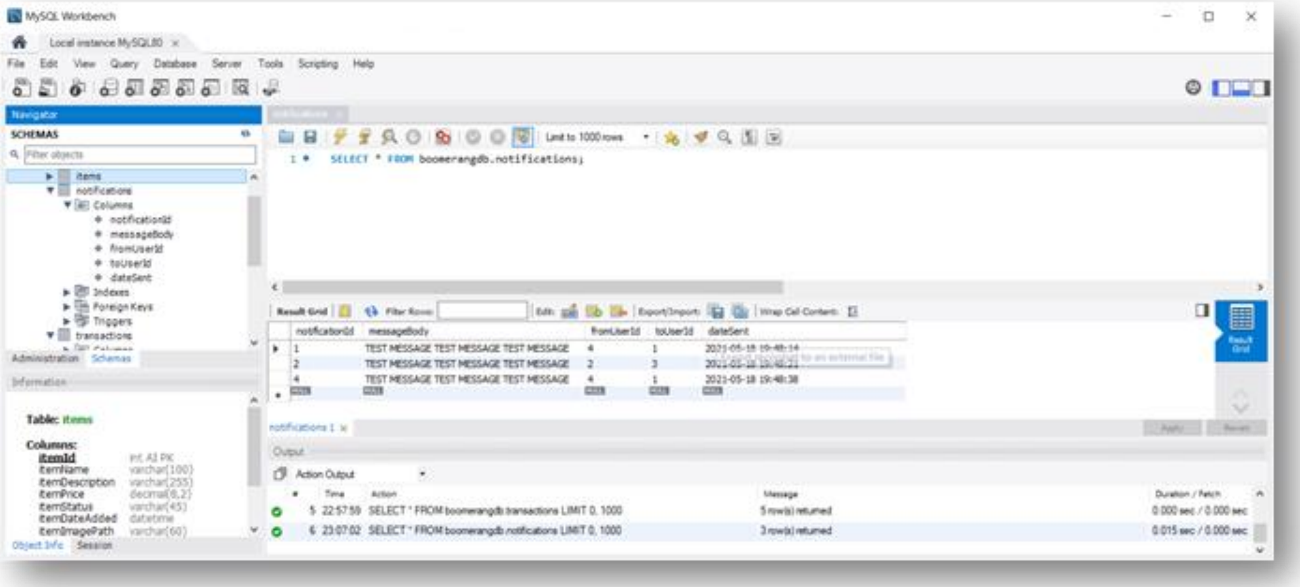
transactions 1

Output

Action Output

```
  #  Time      Action                                             Message              Duration / Fetch
  4  22:48:32  SELECT * FROM boomerangdb.items LIMIT 0, 1000         9 row(s) returned  0.000 sec / 0.000 sec
  5  22:57:59  SELECT * FROM boomerangdb.transactions LIMIT 0, 1000  5 row(s) returned  0.000 sec / 0.000 sec
```

Transactions Table Data Structure



```
1 •  SELECT * FROM boomerangdb.notifications;
```

	notificationId	messageBody	fromUserId	toUserId	dateSent
▶	1	TEST MESSAGE TEST MESSAGE TEST MESSAGE	4	1	2021-05-18 19:48:14
	2	TEST MESSAGE TEST MESSAGE TEST MESSAGE	2	3	2021-05-18 19:48:21
	4	TEST MESSAGE TEST MESSAGE TEST MESSAGE	4	1	2021-05-18 19:48:38
●	NULL	NULL	NULL	NULL	NULL

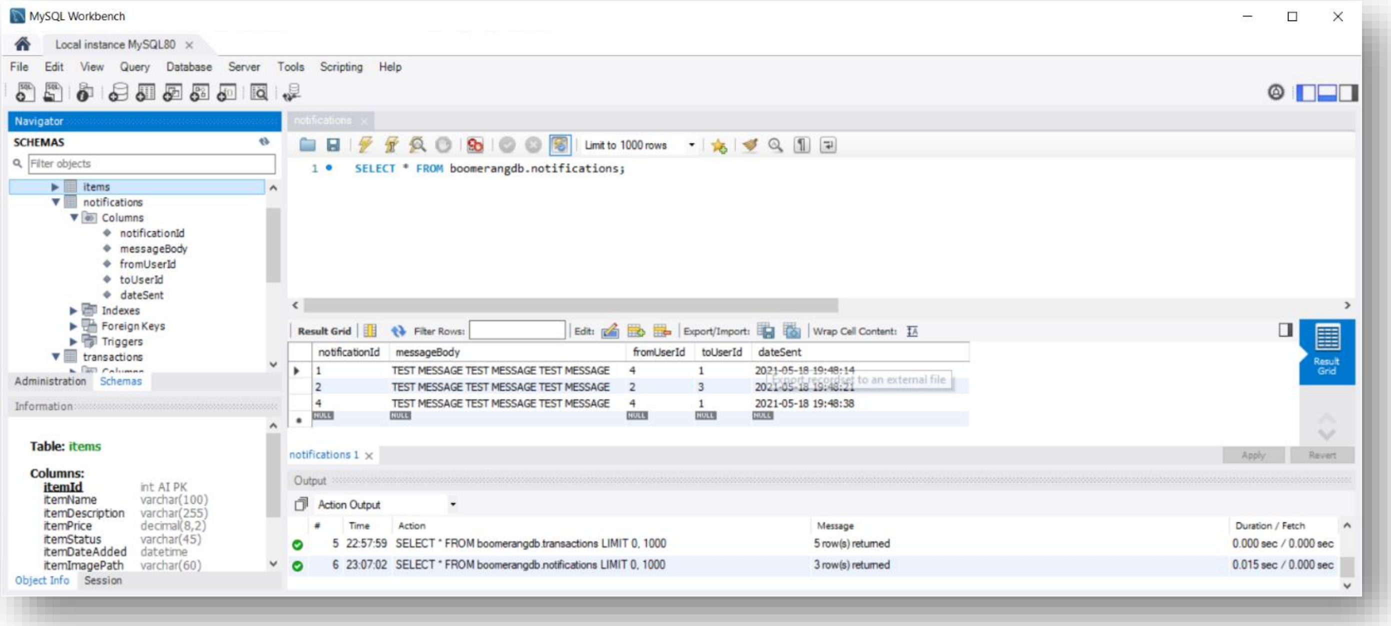
Table: items

Columns:

itemId	int AI PK
itemName	varchar(100)
itemDescription	varchar(255)
itemPrice	decimal(8,2)
itemStatus	varchar(45)
itemDateAdded	datetime
itemImagePath	varchar(60)

	#	Time	Action	Message	Duration / Fetch
✓	5	22:57:59	SELECT * FROM boomerangdb.transactions LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
✓	6	23:07:02	SELECT * FROM boomerangdb.notifications LIMIT 0, 1000	3 row(s) returned	0.015 sec / 0.000 sec

Notifications Table Data Structure



```
1 •  SELECT * FROM boomerangdb.notifications;
```

	notificationId	messageBody	fromUserId	toUserId	dateSent
▶	1	TEST MESSAGE TEST MESSAGE TEST MESSAGE	4	1	2021-05-18 19:48:14
	2	TEST MESSAGE TEST MESSAGE TEST MESSAGE	2	3	2021-05-18 19:48:21
	4	TEST MESSAGE TEST MESSAGE TEST MESSAGE	4	1	2021-05-18 19:48:38
●	NULL	NULL	NULL	NULL	NULL

Table: items

Columns:

itemId	int AI PK
itemName	varchar(100)
itemDescription	varchar(255)
itemPrice	decimal(8,2)
itemStatus	varchar(45)
itemDateAdded	datetime
itemImagePath	varchar(60)

	#	Time	Action	Message	Duration / Fetch
✓	5	22:57:59	SELECT * FROM boomerangdb.transactions LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
✓	6	23:07:02	SELECT * FROM boomerangdb.notifications LIMIT 0, 1000	3 row(s) returned	0.015 sec / 0.000 sec

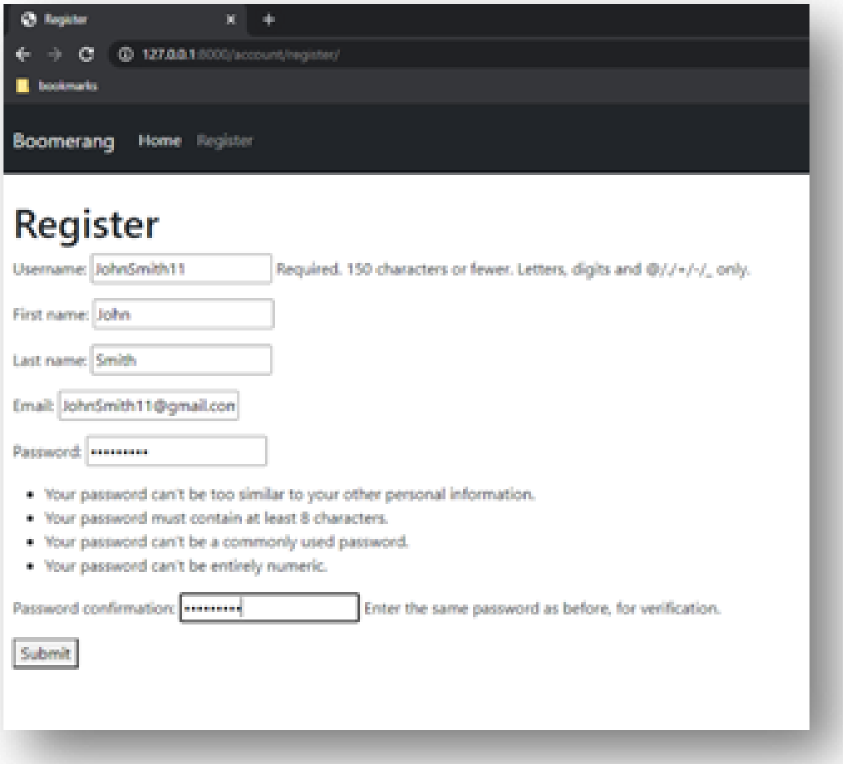
Django framework – User authentication and registration mechanism

The Django framework offers some building blocks that can be used as a foundation for an authentication mechanism for a web application.

Note:
The work currently done in Django does not reflect the visual and colour schemes of the designed Boomerang website, the work currently done is deemed to be a foundation on which further development would be required to integrate it with the other components of the system, namely the MySQL database and the HTML artifacts created.

The Django framework also includes functionality to assist with the development of a “user registration” functionality. Image 1 shows a snippet of the interface that allows user to register on the website.

Image 1 - Shows a basic user registration interface being used.



Register

Username: JohnSmith11 Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name: John

Last name: Smith

Email: JohnSmith11@gmail.con

Password: ••••••••••

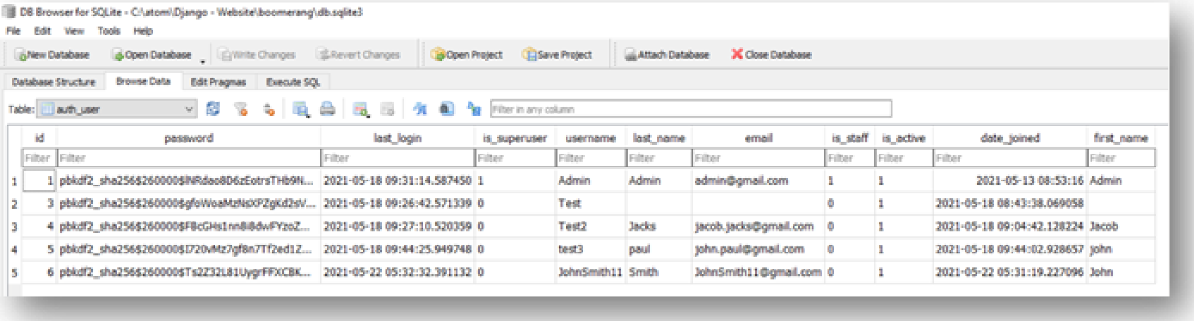
- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation: •••••••••• Enter the same password as before, for verification.

Submit

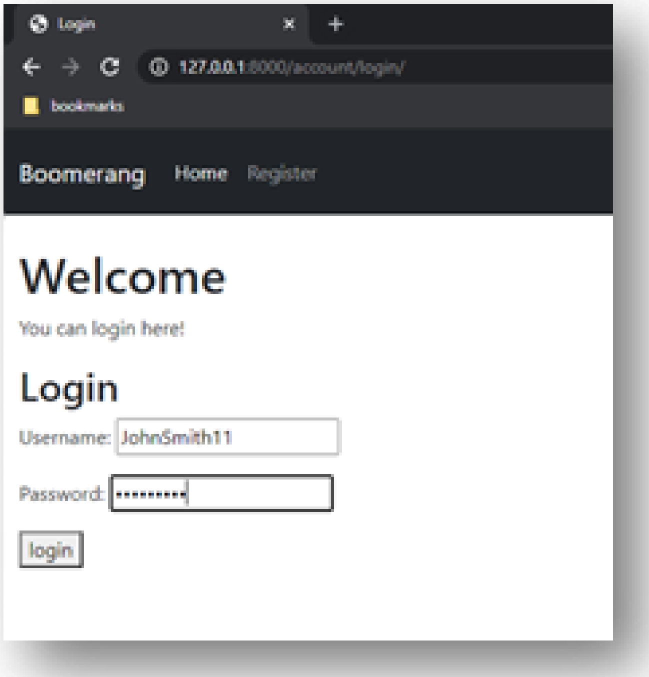
The Django framework comes by default built with an internal SQL Lite database. Currently the user authentication data and the other system data stored in the MySQL database are not integrated.

Image 2 – Illustrates a snippet of the user authentication data stored in Django's SQL Lite database.



	id	password	last_login	is_superuser	username	last_name	email	is_staff	is_active	date_joined	first_name
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	pbkdf2_sha256$260000$lNRdao8D6zEotrsTHb9N...	2021-05-18 09:31:14.587450	1	Admin	Admin	admin@gmail.com	1	1	2021-05-13 08:53:16	Admin
2	3	pbkdf2_sha256$260000$gfoWoaMzNsXPZgKd2sV...	2021-05-18 09:26:42.571339	0	Test			0	1	2021-05-18 08:43:38.069058	
3	4	pbkdf2_sha256$260000$F8cGHs1nn8i8dwFYzoZ...	2021-05-18 09:27:10.520359	0	Test2	Jacks	jacob.jacks@gmail.com	0	1	2021-05-18 09:04:42.128224	Jacob
4	5	pbkdf2_sha256$260000$I720vMz7gf8n7Tf2ed1Z...	2021-05-18 09:44:25.949748	0	test3	paul	john.paul@gmail.com	0	1	2021-05-18 09:44:02.928657	john
5	6	pbkdf2_sha256$260000$Ts2Z32L81UygrFFXCBK...	2021-05-22 05:32:32.391132	0	JohnSmith11	Smith	JohnSmith11@gmail.com	0	1	2021-05-22 05:31:19.227096	John

Image 3 – Demonstrates a user login interface being used.



Welcome

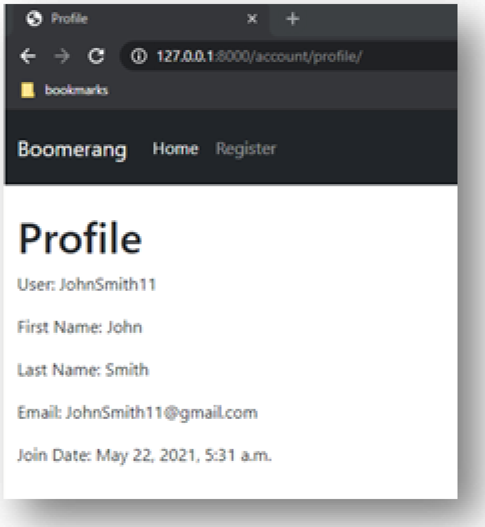
You can login here!

Login

Username: JohnSmith11

Password: ••••••••••

Image 4 - Once logged in, the user can be presented with a summary showing their personal details.



Profile

User: JohnSmith11

First Name: John

Last Name: Smith

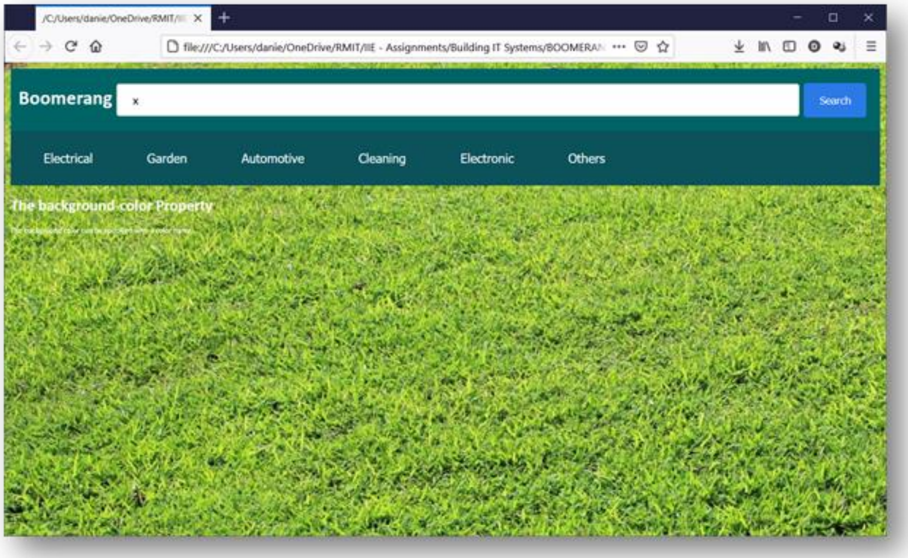
Email: JohnSmith11@gmail.com

Join Date: May 22, 2021, 5:31 a.m.

Boomerang homepage – Template HTML artifact

The Boomerang homepage, including its colour scheme and branding has been designed as a template for future iterations and additional pages of the website.

Currently this page is “static” and not integrated with the Django framework or the MySQL database.


/C:/Users/danie/OneDrive/RMIT/IIE
file:///C:/Users/danie/OneDrive/RMIT/IIE - Assignments/Building IT Systems/BOOMERAN
Boomerang
Search
Electrical
Garden
Automotive
Cleaning
Electronic
Others
The background-color Property
The background color can be specified with a color name

2.3. Scope Creep

Initially when designing the scope of the project we had 5 Minimal Viable Features (MVFs) in mind that we wanted to create for the Web application. These being; User Profile, Search Engine, Goods and Service Listings, Messaging and Communication system, and Ability to add items to be leaser by others.

We believe we did not steer away or far from our initial scope when designing and creating artefacts in correlation to our 5 MVFs. Our high fidelity Figma prototype displays a fully function and clickable model that displays all our 5 MVFs we planned to create. We also designed diagrams with LucidChart, demonstrating these 5 MVFs without steering away from the original ideas in mind and adding new unwarranted concepts.

This is also exhibited through the completed code in the Django Framework for the Login and Registration form. This Registration form allows users to make accounts for our website and be able to have a personal login. When they login the consumer is able to see their own User Profile when they click on their profile. When a user has created the account, it is stored within our SQLite Database which is connected to the Django Framework. This is so their account details are saved and they are able to login at any time.

Also, we created a SQL Database that illustrates how the user will have the ability to add items and be able to give full detailed descriptions of their listing to others who want to lease their product. We believe the features that we have created, do not steer away from our original plan and design concepts when designing our Web Application. Hence, as a group we believe we had little to no Scope creep present during the outcome of our Web application to date.

2.4. Progress

The team has made excellent progress throughout the project in terms of meeting deliverables and completing reporting. During the early stages of the assignment, the team was able to conduct extensive research and planning for development, which greatly aided us in keeping the project on track and on time.

Planning – When planning out the project, we identified each of the key features that we wanted to implement and developed a comprehensive step-by-step plan for each of them. Based on this, we estimated the amount of time required for each phase and distributed the work for each feature across the time available for project development. We attempted to distribute the workload evenly across the weeks, and across the various workstreams.
We completed the necessary tasks during the first few weeks of project creation (Assignment and individual task). The first four weeks were spent forming groups,

collecting, and creating project ideas, as well as proposing and documenting project ideas. The group concentrated on the details of the project and spend time understanding and explore the idea. Individuals researched the technologies and resources required for the project. Because it is critical to have an estimate for any project, the group agreed to allocate a few hours on Trello every week. Trello assisted us in keeping track of our weekly activities.

We discovered that some of the development tasks required substantially more time than initially anticipated. This was particularly the case for designing and planning the artefacts using LucidChart and Figma. We need to devote few hours to each of the variable features. A significant amount of time was allocated to learning Django and other tools and technologies to understand and experience which would be used in the project's production.

While doing daily work (assignments), the group had to devote a few hours each day to learning about programming languages, tools, and technologies such as Django, Python, and SQL, as we planned to use those platforms to develop our project Boomerang. Due to dependencies between learning and experimenting with Django, Python, and SQL, the start of building the EVFs were delayed. We were all new to these networks as a group, so we eventually decided that implementing EVFs was no longer feasible, but we gave it our all in developing our project.
With the decision and difficulties, we were experiencing, we adjusted the target dates for the MVFs which freed up the team’s time to focus on implementing the EVFs as much as possible. By the end of the available implementation period, we successfully completed a few MVFs and tried implementing EVFs.

Prototype– Most of the group's time was spent on wireframes. It is critical to understand the fundamental concept, and nothing beats planning with diagrams and prototypes. Diagrams and charts help to simplify the steps, so we decided to set aside a few hours each day for two weeks to create a basic layout in LucidChart. We spend adequate time in Figma to deliver high fidelity prototype.

MVF Production

The timeline indicated that the project would be ready for testing and debugging by week 10. We believe we could have built our project if we didn't have the constraints of time. Currently, the team has created a main page with HTML and Notepad++, collected and have a SQL database containing users, administrators, and product information, and have a login & sign-up system and admin system in Django.

It's not about the final product that matters; it's about the lessons learned along the way. Team Alpha was unable to integrate their work into a single board, but as a group, we did not give up and continued to research and find new ways to pile the work. In terms of feature development, the project is on time with planning, designing, creating features using a certain platform as predicted in the first report.

While the team has not yet produced a fully revised schedule, changes to how the team functions have been made in order to present a completed project. The most significant change has resulted from encouraging team members to research and build individual features before bringing the project together. This has given each member increased responsibility, which has aided in the project's advancement.

Overall, Team Alpha generally progressed smoothly throughout the duration of the project (includes all the assignments) with only a few roadblocks caused by the group's inexperience with the platform – Django, Python, Programming Language, and developing IT Systems.
A variety of features were modified to fit the timeline based on additional research and roadblocks with the team following methods outlined in the Agile Ways of Working Micro credential, as outlined in other parts of the study. The team prioritised the values and principles outlined in this course, such as our adaptability when it comes to MVF changes; for example, database architecture was well-scoped, but certain aspects of testing and validation had to be modified. Since the team's lack of experience designing and constructing IT systems (using these platforms and coding and software barriers) ensured accuracy around scoping elements of the project.

As report content was developed gradually, the team also prioritised the value “working software is the primary measure of progress,” and actual product development was prioritised as well. Learning something new requires time therefore we decided to allocate some time to learn and experiment with all the technologies which are being used in the project. Prioritising the project (wireframes, learning, and testing the platform) over documentation allowed for a more adaptable approach to teamwork.

Consistent team meetings, both with our mentor and with our group, were well planned, allowing us to discuss any obstacles without deviating from the main topic. Reflecting at key points during these meetings was also essential to the reassessment of MVFs.
This organisation provided the team with the opportunity to gradually develop aspects of the project without having to scramble at the last minute.

Testing

One of the components that the team placed significant emphasis on, was the testing of whether the relational data required for the web application was stored in a robust way, and in a form that would avoid data corruption or data loss.

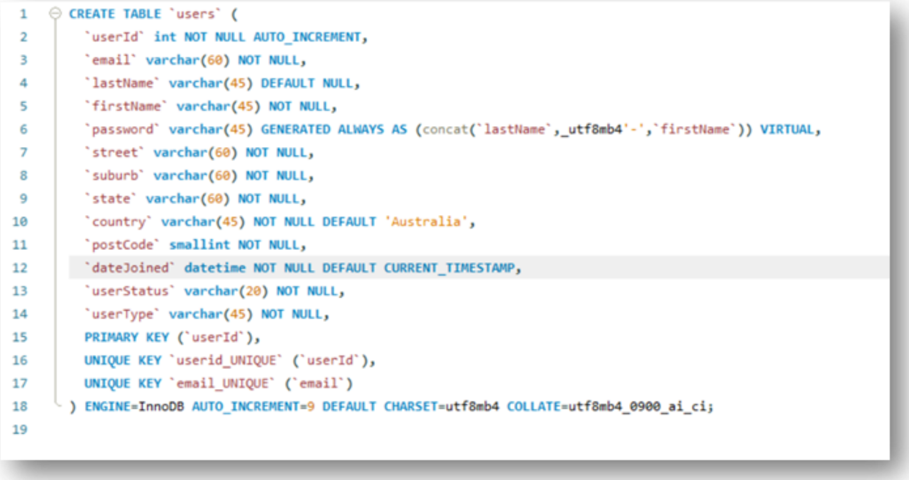
Throughout the development of the data database and its schema, several methods were researched to help us implement a consistent data store architecture. Some of the methods used include using appropriate data types for storing different types of information, defining mandatory fields, defining default values where applicable, defining unique primary keys and most importantly, configuring data constraints via the use of various Foreign Key relations.

All the tables are related to one another via indexed and foreign keys, this ensure the consistency of the data is always maintained. The table constraints also stop data from being deleted when the data is referenced in other system tables.

Testing of the user data storage

The schema of this table is designed to only accept the correct data types for each one of its fields.
Some of the fields are “generated fields, an example of this is the “password” field.
Some of the fields also have default values assigned.

Schema code developed for the “users” table



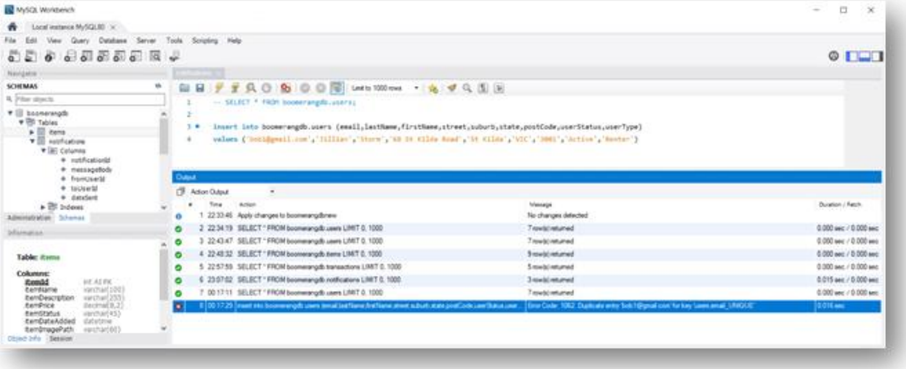
```
CREATE TABLE `users` (
  `userId` int NOT NULL AUTO_INCREMENT,
  `email` varchar(60) NOT NULL,
  `lastName` varchar(45) DEFAULT NULL,
  `firstName` varchar(45) NOT NULL,
  `password` varchar(45) GENERATED ALWAYS AS (concat(`lastName`,_utf8mb4'-',`firstName`)) VIRTUAL,
  `street` varchar(60) NOT NULL,
  `suburb` varchar(60) NOT NULL,
  `state` varchar(60) NOT NULL,
  `country` varchar(45) NOT NULL DEFAULT 'Australia',
  `postCode` smallint NOT NULL,
  `dateJoined` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `userStatus` varchar(20) NOT NULL,
  `userType` varchar(45) NOT NULL,
  PRIMARY KEY (`userId`),
  UNIQUE KEY `userid_UNIQUE` (`userId`),
  UNIQUE KEY `email_UNIQUE` (`email`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Testing:

1) When adding new users to the system, the system will not accept any data which contain fields that are marked as *unique*. In the Image 1 below, the database rejects a user being created with an email which already exists on the system.

The database outputs the error:
"… 00:17:29 insert into boomerangdb.users
Error Code: 1062. Duplicate entry 'bob1@gmail.com' for key 'users.email_UNIQUE'
0.016 sec …"

Image 1



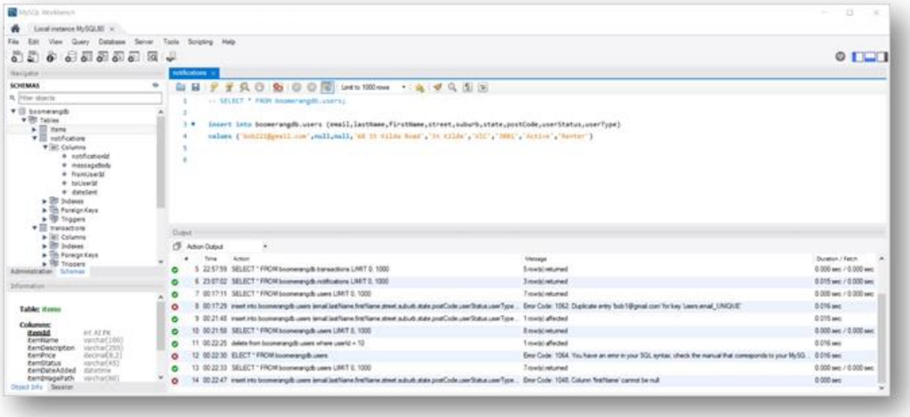
```
-- SELECT * FROM boomerangdb.users;

insert into boomerangdb.users (email,lastName,firstName,street,suburb,state,postCode,userStatus,userType)
values ('bob1@gmail.com','Jillian','Storm','68 St Kilda Road','St Kilda','VIC','3001','Active','Renter')
```

```
#  Time      Action                                                    Message                                                                    Duration / Fetch
1  22:33:46  Apply changes to boomerangdbnew                           No changes detected
2  22:34:19  SELECT * FROM boomerangdb.users LIMIT 0, 1000             7 row(s) returned                                                          0.000 sec / 0.000 sec
3  22:43:47  SELECT * FROM boomerangdb.users LIMIT 0, 1000             7 row(s) returned                                                          0.000 sec / 0.000 sec
4  22:48:32  SELECT * FROM boomerangdb.items LIMIT 0, 1000             9 row(s) returned                                                          0.000 sec / 0.000 sec
5  22:57:59  SELECT * FROM boomerangdb.transactions LIMIT 0, 1000      5 row(s) returned                                                          0.000 sec / 0.000 sec
6  23:07:02  SELECT * FROM boomerangdb.notifications LIMIT 0, 1000     3 row(s) returned                                                          0.015 sec / 0.000 sec
7  00:17:11  SELECT * FROM boomerangdb.users LIMIT 0, 1000             7 row(s) returned                                                          0.000 sec / 0.000 sec
8  00:17:29  insert into boomerangdb.users (email,lastName,firstName,street,suburb,state,postCode,userStatus,user...  Error Code: 1062. Duplicate entry 'bob1@gmail.com' for key 'users.email_UNIQUE'  0.016 sec
```

2) The user also does not accept the creating of new users where mandatory fields are missing. In the Image 2 below, an attempt is made to create a user without firstName and lastName. The database throws an error.

Image 2



```
-- SELECT * FROM boomerangdb.users;

insert into boomerangdb.users (email,lastName,firstName,street,suburb,state,postCode,userStatus,userType)
values ('bob221@gmail.com',null,null,'68 St Kilda Road','St Kilda','VIC','3001','Active','Renter')
```

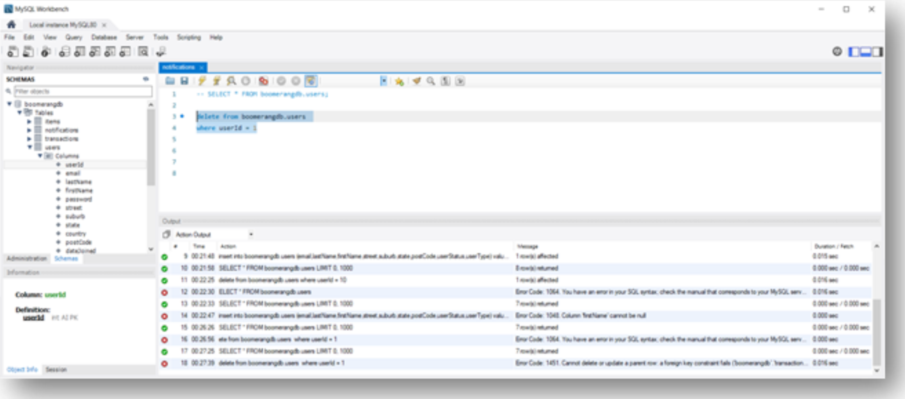
```
#   Time      Action                                                    Message                                                                    Duration / Fetch
5   22:57:59  SELECT * FROM boomerangdb.transactions LIMIT 0, 1000      5 row(s) returned                                                          0.000 sec / 0.000 sec
6   23:07:02  SELECT * FROM boomerangdb.notifications LIMIT 0, 1000     3 row(s) returned                                                          0.015 sec / 0.000 sec
7   00:17:11  SELECT * FROM boomerangdb.users LIMIT 0, 1000             7 row(s) returned                                                          0.000 sec / 0.000 sec
8   00:17:29  insert into boomerangdb.users (email,lastName,firstName,street,suburb,state,postCode,userStatus,userType...  Error Code: 1062. Duplicate entry 'bob1@gmail.com' for key 'users.email_UNIQUE'  0.016 sec
9   00:21:45  insert into boomerangdb.users (email,lastName,firstName,street,suburb,state,postCode,userStatus,userType...  1 row(s) affected  0.015 sec
10  00:21:58  SELECT * FROM boomerangdb.users LIMIT 0, 1000             8 row(s) returned                                                          0.000 sec / 0.000 sec
11  00:22:25  delete from boomerangdb.users where userId = 10           1 row(s) affected                                                          0.016 sec
12  00:22:30  ELECT * FROM boomerangdb.users                            Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL...  0.016 sec
13  00:22:33  SELECT * FROM boomerangdb.users LIMIT 0, 1000             7 row(s) returned                                                          0.000 sec / 0.000 sec
14  00:22:47  insert into boomerangdb.users (email,lastName,firstName,street,suburb,state,postCode,userStatus,userType...  Error Code: 1048. Column 'firstName' cannot be null  0.000 sec
```

The database outputs the error:

"… 00:22:47 insert into boomerangdb.users \Error Code: 1048. Column 'firstName' cannot be null 0.000 sec …"

3) Another important user test is, to attempt to delete a user from the system when the user is referenced in other system tables. The Image 3 below shows an attempt to delete the user with the userId 1

Image 3



```
-- SELECT * FROM boomerangdb.users;

delete from boomerangdb.users
where userId = 1
```

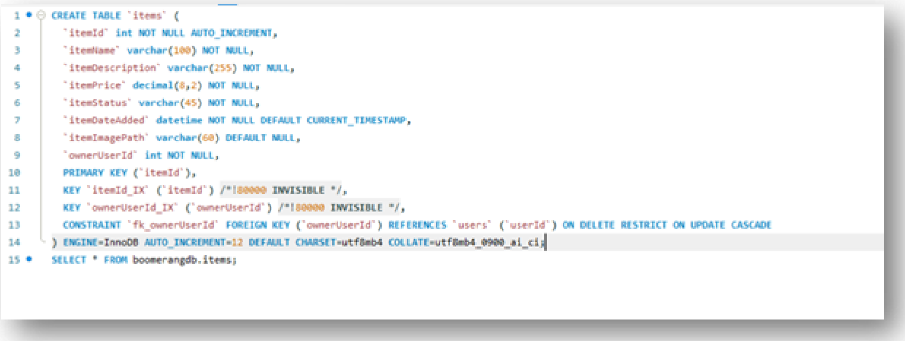
```
#   Time      Action                                                                                          Message                                                                              Duration / Fetch
9   00:21:48  insert into boomerangdb.users (email,lastName,firstName,street,suburb,state,postCode,userStatus,userType) valu...  1 row(s) affected                                                                    0.015 sec
10  00:21:58  SELECT * FROM boomerangdb.users LIMIT 0, 1000                                                   8 row(s) returned                                                                    0.000 sec / 0.000 sec
11  00:22:25  delete from boomerangdb.users where userId = 10                                                 1 row(s) affected                                                                    0.016 sec
12  00:22:30  ELECT * FROM boomerangdb.users                                                                  Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL serv...  0.016 sec
13  00:22:33  SELECT * FROM boomerangdb.users LIMIT 0, 1000                                                   7 row(s) returned                                                                    0.000 sec / 0.000 sec
14  00:22:47  insert into boomerangdb.users (email,lastName,firstName,street,suburb,state,postCode,userStatus,userType) valu...  Error Code: 1048. Column 'firstName' cannot be null                                  0.000 sec
15  00:26:26  SELECT * FROM boomerangdb.users LIMIT 0, 1000                                                   7 row(s) returned                                                                    0.000 sec / 0.000 sec
16  00:26:56  ete from boomerangdb.users  where userId = 1                                                    Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL serv...  0.000 sec
17  00:27:25  SELECT * FROM boomerangdb.users LIMIT 0, 1000                                                   7 row(s) returned                                                                    0.000 sec / 0.000 sec
18  00:27:39  delete from boomerangdb.users  where userId = 1                                                 Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`boomerangdb`.`transaction...  0.016 sec
```

The database outputs the error:
"… 00:27:39 delete from boomerangdb.users where userId = 1   Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`boomerangdb`.`transactions`, CONSTRAINT `fk_lessorID` FOREIGN KEY (`lessorId`) 0.016 sec …"

Testing of the items data storage

On the “items” table, the ownerUserId column is a foreign key assigned to the userID on the table “users”. This stops items from being created and being associated to non-existing users. This table contains information including item name, description, price, status, data added and who the userId of the owner.

Schema code developed for this table

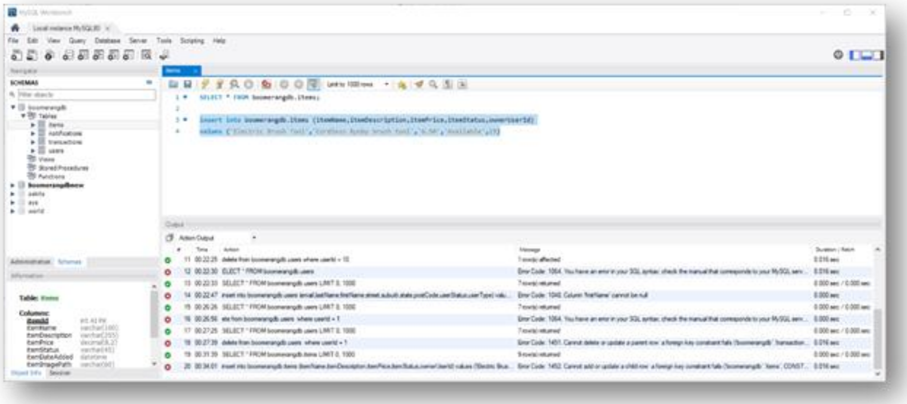


```
CREATE TABLE `items` (
  `itemId` int NOT NULL AUTO_INCREMENT,
  `itemName` varchar(100) NOT NULL,
  `itemDescription` varchar(255) NOT NULL,
  `itemPrice` decimal(8,2) NOT NULL,
  `itemStatus` varchar(45) NOT NULL,
  `itemDateAdded` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `itemImagePath` varchar(60) DEFAULT NULL,
  `ownerUserId` int NOT NULL,
  PRIMARY KEY (`itemId`),
  KEY `itemId_IX` (`itemId`) /*!80000 INVISIBLE */,
  KEY `ownerUserId_IX` (`ownerUserId`) /*!80000 INVISIBLE */,
  CONSTRAINT `fk_ownerUserId` FOREIGN KEY (`ownerUserId`) REFERENCES `users` (`userId`) ON DELETE RESTRICT ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
SELECT * FROM boomerangdb.items;
```

Testing:

1) The database will not accept the creation of a new item where the userId is invalid, the Foreign key constraints stop that from happening. Refer to Image 4

Image 4



```
SELECT * FROM boomerangdb.items;

insert into boomerangdb.items (itemName,itemDescription,itemPrice,itemStatus,ownerUserId)
values ('Electric Brush Tool','Cordless Ryoby brush tool','6.50','Available',19)
```

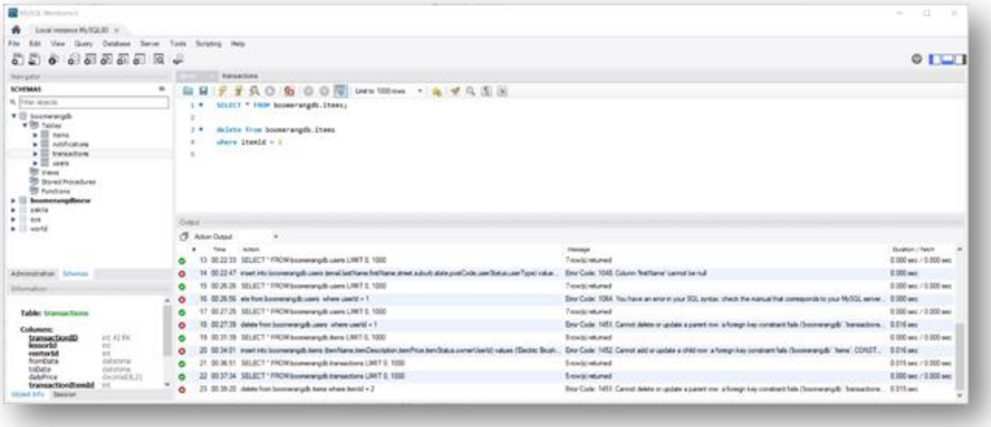
```
#   Time      Action                                                        Message                                                                          Duration / Fetch
11  00:22:25  delete from boomerangdb.users where userId = 10               1 row(s) affected                                                                0.016 sec
12  00:22:30  ELECT * FROM boomerangdb.users                                Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL serv...  0.016 sec
13  00:22:33  SELECT * FROM boomerangdb.users LIMIT 0, 1000                 7 row(s) returned                                                                0.000 sec / 0.000 sec
14  00:22:47  insert into boomerangdb.users (email,lastName,firstName,street,suburb,state,postCode,userStatus,userType) valu...  Error Code: 1048. Column 'firstName' cannot be null  0.000 sec
15  00:26:26  SELECT * FROM boomerangdb.users LIMIT 0, 1000                 7 row(s) returned                                                                0.000 sec / 0.000 sec
16  00:26:56  ete from boomerangdb.users where userId = 1                   Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL serv...  0.000 sec
17  00:27:25  SELECT * FROM boomerangdb.users LIMIT 0, 1000                 7 row(s) returned                                                                0.000 sec / 0.000 sec
18  00:27:39  delete from boomerangdb.users where userId = 1                Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`boomerangdb`.`transaction...  0.016 sec
19  00:31:39  SELECT * FROM boomerangdb.items LIMIT 0, 1000                 9 row(s) returned                                                                0.000 sec / 0.000 sec
20  00:34:01  insert into boomerangdb.items (itemName,itemDescription,itemPrice,itemStatus,ownerUserId) values ('Electric Bru...  Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`boomerangdb`.`items`, CONST...  0.016 sec
```

The database outputs the error:

“… 00:34:01 insert into boomerangdb.items Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`boomerangdb`.`items`, CONSTRAINT `fk_ownerUserId` FOREIGN KEY (`ownerUserId`) 0.016 sec…”

2) The database will also not accept the deletion of items that are referenced in other tables. In this example show in Image 5, an attempt is made to delete the "itemId 2".

Image 5



```
SELECT * FROM boomerangdb.items;

delete from boomerangdb.items
where itemId = 2

```

```
    Time      Action                                                                                          Message                                                                                                  Duration / Fetch
13  00:22:33  SELECT * FROM boomerangdb.users LIMIT 0, 1000                                                   7 row(s) returned                                                                                        0.000 sec / 0.000 sec
14  00:22:47  insert into boomerangdb.users (email,lastName,firstName,street,suburb,state,postCode,userStatus,userType) value...  Error Code: 1048. Column 'firstName' cannot be null                                  0.000 sec
15  00:26:26  SELECT * FROM boomerangdb.users LIMIT 0, 1000                                                   7 row(s) returned                                                                                        0.000 sec / 0.000 sec
16  00:26:56  ete from boomerangdb.users where userId = 1                                                     Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server...  0.000 sec
17  00:27:25  SELECT * FROM boomerangdb.users LIMIT 0, 1000                                                   7 row(s) returned                                                                                        0.000 sec / 0.000 sec
18  00:27:35  delete from boomerangdb.users where userId = 1                                                  Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`boomerangdb`.`transactions...  0.016 sec
19  00:31:39  SELECT * FROM boomerangdb.items LIMIT 0, 1000                                                   9 row(s) returned                                                                                        0.000 sec / 0.000 sec
20  00:34:01  insert into boomerangdb.items (itemName,itemDescription,itemPrice,itemStatus,ownerUserId) values ('Electric Brush...  Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`boomerangdb`.`items`, CONST...  0.016 sec
21  00:36:51  SELECT * FROM boomerangdb.transactions LIMIT 0, 1000                                            5 row(s) returned                                                                                        0.015 sec / 0.000 sec
22  00:37:34  SELECT * FROM boomerangdb.transactions LIMIT 0, 1000                                            5 row(s) returned                                                                                        0.000 sec / 0.000 sec
23  00:39:20  delete from boomerangdb.items where itemId = 2                                                  Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`boomerangdb`.`transactions...  0.015 sec
```

The database outputs the error:

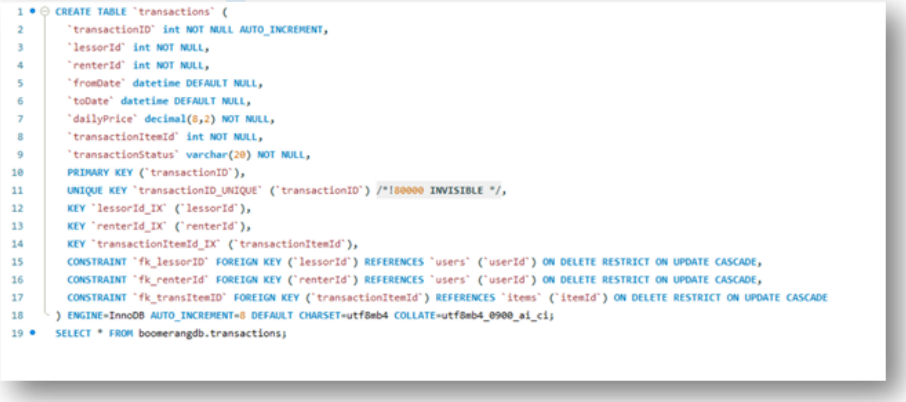
"… 00:39:20 delete from boomerangdb.items where itemId = 2 Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`boomerangdb`.`transactions`, CONSTRAINT `fk_transItemID` FOREIGN KEY (`transactionItemId`) 0.015 sec…"

Testing of the transactions data storage

On the “transactions” table, the renterId and lessorId columns are both Foreign Keys assigned to the userID on the table “users”. This is designed to avoid transactions from becoming invalid due to non-existing users.

There is an additional Foreign Key on this table on the column transactionItemId, this foreign key is assigned to the itemId on the “items” table, this avoids transactions with invalid items on it.

Schema code developed for this table

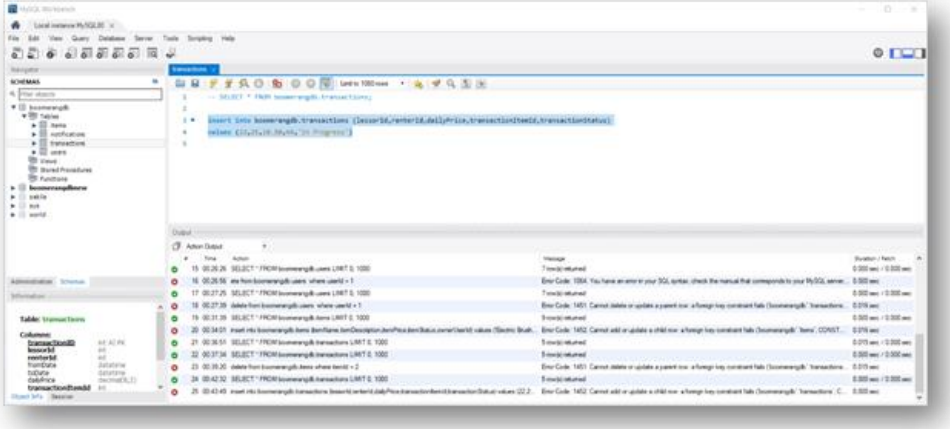


```sql
CREATE TABLE `transactions` (
  `transactionID` int NOT NULL AUTO_INCREMENT,
  `lessorId` int NOT NULL,
  `renterId` int NOT NULL,
  `fromDate` datetime DEFAULT NULL,
  `toDate` datetime DEFAULT NULL,
  `dailyPrice` decimal(8,2) NOT NULL,
  `transactionItemId` int NOT NULL,
  `transactionStatus` varchar(20) NOT NULL,
  PRIMARY KEY (`transactionID`),
  UNIQUE KEY `transactionID_UNIQUE` (`transactionID`) /*!80000 INVISIBLE */,
  KEY `lessorId_IX` (`lessorId`),
  KEY `renterId_IX` (`renterId`),
  KEY `transactionItemId_IX` (`transactionItemId`),
  CONSTRAINT `fk_lessorID` FOREIGN KEY (`lessorId`) REFERENCES `users` (`userId`) ON DELETE RESTRICT ON UPDATE CASCADE,
  CONSTRAINT `fk_renterId` FOREIGN KEY (`renterId`) REFERENCES `users` (`userId`) ON DELETE RESTRICT ON UPDATE CASCADE,
  CONSTRAINT `fk_transItemID` FOREIGN KEY (`transactionItemId`) REFERENCES `items` (`itemId`) ON DELETE RESTRICT ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
SELECT * FROM boomerangdb.transactions;
```

Testing:

1) The database will not accept transactions to be create where the “lessorId”, “renterId”, and “ItemId” are invalid Ids. This ensures a robust level of data integrity for the system. The Image 6 below shows an attempt to create an invalid transaction

Image 6



```sql
-- SELECT * FROM boomerangdb.transactions;

insert into boomerangdb.transactions (lessorId,renterId,dailyPrice,transactionItemId,transactionStatus)
values ([illegible],'In Progress')
```

```
#   Time      Action                                                          Message                                                                                      Duration / Fetch
15  00:26:26  SELECT * FROM boomerangdb.users LIMIT 0, 1000                   7 row(s) returned                                                                            0.000 sec / 0.000 sec
16  00:26:56  ete from boomerangdb.users where userId = 1                     Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server...   0.000 sec
17  00:27:25  SELECT * FROM boomerangdb.users LIMIT 0, 1000                   7 row(s) returned                                                                            0.000 sec / 0.000 sec
18  00:27:39  delete from boomerangdb.users where userId = 1                  Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`boomerangdb`.`transactions`...   0.016 sec
19  00:31:39  SELECT * FROM boomerangdb.items LIMIT 0, 1000                   9 row(s) returned                                                                            0.000 sec / 0.000 sec
20  00:34:01  insert into boomerangdb.items (itemName,itemDescription,itemPrice,itemStatus,ownerUserId) values ('Electric Brush...   Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`boomerangdb`.`items`, CONST...   0.016 sec
21  00:36:51  SELECT * FROM boomerangdb.transactions LIMIT 0, 1000            5 row(s) returned                                                                            0.015 sec / 0.000 sec
22  00:37:34  SELECT * FROM boomerangdb.transactions LIMIT 0, 1000            5 row(s) returned                                                                            0.000 sec / 0.000 sec
23  00:39:20  delete from boomerangdb.items where itemId = 2                  Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`boomerangdb`.`transactions`...   0.015 sec
24  00:42:32  SELECT * FROM boomerangdb.transactions LIMIT 0, 1000            5 row(s) returned                                                                            0.000 sec / 0.000 sec
25  00:43:49  insert into boomerangdb.transactions (lessorId,renterId,dailyPrice,transactionItemId,transactionStatus) values (22,2...   Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`boomerangdb`.`transactions`, C...   0.000 sec
```

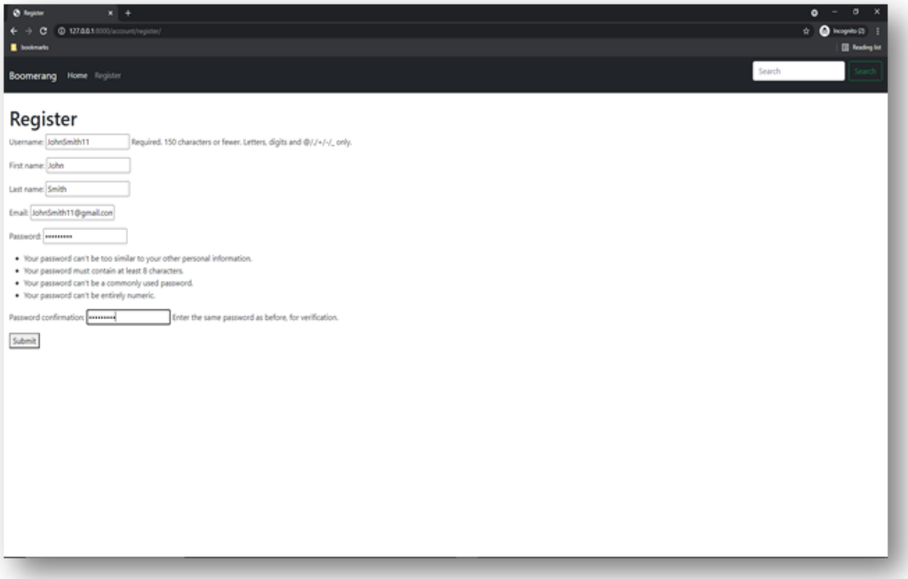
The database outputs the error:

“… 00:43:49 insert into boomerangdb.transactions	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`boomerangdb`.`transactions`, CONSTRAINT `fk_lessorID` FOREIGN KEY (`lessorId`) 0.000 sec…”

Testing of the user Django framework authentication mechanism

1) To ensure that the registration and login features are in working order, we made sure to test it using a dummy user as demonstrated in the example below. To test the registration and login of our Web Application, we created a new User for a consumer by the name of 'John Smith'.

Image 1



Register

Username: JohnSmith11 Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name: John

Last name: Smith

Email: JohnSmith11@gmail.con

Password: ••••••••••

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation: •••••••••• Enter the same password as before, for verification.

Submit

After John has successfully created the login, it is automatically stored in our SQLite Database which is attached to our Django Framework. This is so John or any user can easily login with their Username and Password later.

Image 2

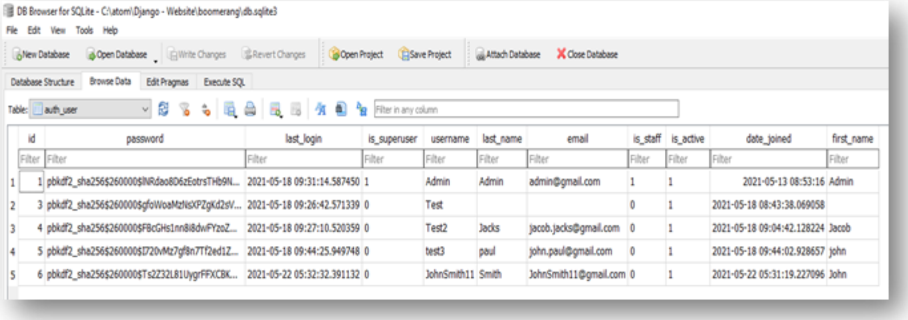
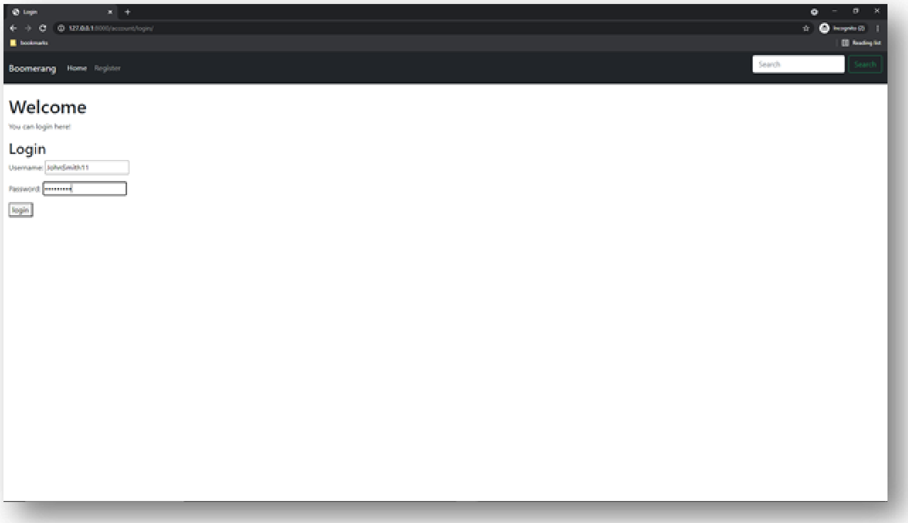


Table: auth_user

	id	password	last_login	is_superuser	username	last_name	email	is_staff	is_active	date_joined	first_name
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	pbkdf2_sha256$260000$lNRdao8D6zEotrsTHb9N...	2021-05-18 09:31:14.587450	1	Admin	Admin	admin@gmail.com	1	1	2021-05-13 08:53:16	Admin
2	3	pbkdf2_sha256$260000$gfoWoaMzNsXPZgKd2sV...	2021-05-18 09:26:42.571339	0	Test			0	1	2021-05-18 08:43:38.069058	
3	4	pbkdf2_sha256$260000$FBcGHs1nn8i8dwFYzoZ...	2021-05-18 09:27:10.520359	0	Test2	Jacks	jacob.jacks@gmail.com	0	1	2021-05-18 09:04:42.128224	Jacob
4	5	pbkdf2_sha256$260000$l720vMz7gf8n7Tf2ed1Z...	2021-05-18 09:44:25.949748	0	test3	paul	john.paul@gmail.com	0	1	2021-05-18 09:44:02.928657	john
5	6	pbkdf2_sha256$260000$Ts2Z32L81UygrFFXCBK...	2021-05-22 05:32:32.391132	0	JohnSmith11	Smith	JohnSmith11@gmail.com	0	1	2021-05-22 05:31:19.227096	John

Now that John has Registered to the website, he can easily login from the login page at any time.

Image 3



Welcome

You can login here!

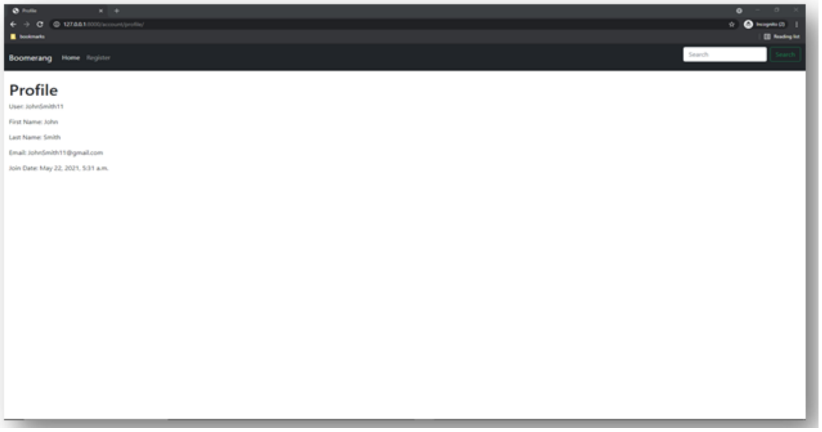
Login

Username: JohnSmith11

Password: ••••••••••

John is now logged in and the website knows what User is logged in therefore, John can see his details he used when creating the account.

Image 4



Profile

User: JohnSmith11

First Name: John

Last Name: Smith

Email: JohnSmith11@gmail.com

Join Date: May 22, 2021, 5:31 a.m.

This demonstrates that the login feature is operational, as ‘John Smith’ was successfully able to create an account. This is beneficial as it can be used later with other factors of the website, like when a user wants to inquire about an item or when they want to list an item it can be registered to the user that is currently logged in at the time.

3. Tools and Technologies

Collaboration Tools	
Microsoft Teams	Microsoft Teams is a team collaboration tool that helps you to keep the whole team in one place and collaborate with them. Since team members can enter existing networks or create their own, we used Teams. Inside networks, members may have on-the-spot meetings, talk, and share files. Members of the team would be able to see what is scheduled for the day or week. Alternatively, call a meeting. With Activity, members will catch up on all their unread messages, @mentions, replies, and more. For the team, productivity will rise, and teamwork will improve. Transparency has improved - For several companies, keeping everyone in the loop and making everyone feel like they know what's going on is a huge challenge. https://teams.microsoft.com/l/team/19%3ad8865f510967487ca4a8002334e15b18%40thread.tacv2/conversations?groupId=99c9ac64-a2c2-429d-ad10-17fa6048b144&tenantId=d1323671-cdbe-4417-b4d4-bdb24b51316b
Trello	Trello is a project management and efficiency tool focused on the Kanban technique. The Kanban technique tracks the status (or swim lanes) of discrete deliverables/objectives using notes/cards. Trello will be used to keep track of project deliverables and other projects, as well as to communicate them. Trello allowed team members to quickly see the tasks were allocated to them, which tasks were complete or incomplete, and what actions were due next. Each team member is in charge of reviewing Trello, marking completed tasks, and anticipating the next mission. https://trello.com/b/SG3eax4b/sp1-group14
Rmit Canvas	All of your course announcements and resources, such as class notes and tools, can be found in RMIT's learning management system (lms). Canvas will also be used to participate in class activities, complete assessments, show and submit assignments, and receive grades and feedback. https://rmit.instructure.com/groups/320969
Emails	All RMIT students have access to the RMIT email system. When Team Alpha was having trouble contacting team members, they used email as an alternate communication channel. It was basically troubleshooting with any team members who weren't using Teams. https://outlook.office.com/mail/inbox

Design Tools	
LucidChart	LucidChart is a proprietary web-based platform that enables users to collaborate on chart and diagram development, revision, and sharing. https://lucid.app/lucidchart/5ae21ae9-5d71-4200-9b46-60a1fd551488/edit?page=0_0#
Figma	Figma is an interactive modelling and prototyping tool. It's a user interface and user experience design platform that can be used to make websites, apps, or smaller user interface components that can be incorporated into larger projects. Figma will be used to offer our project shape and structure. We'll keep using Figma because the next assignment can enable us to build a prototype, and because it's an interactive tool, we can also use it to show off our website. https://www.figma.com/proto/vJbJ5IcnhHjUTmPLraDQhx/Boomerang?node-id=1%3A2&scaling=min-zoom&page-id=0%3A1
Wix	Wix is a website-building software that is both free and user-friendly. Our user-friendly technology and powerful built-in features empower our users to create professional websites that look great on any platform. Our Boomarang Logo was modified with the help of Wix. https://www.wix.com/logo/maker
Development Tools	
GitHub	GitHub allows users to quickly access code and work on projects in a timely manner. It's a free source control and code repository. Rather than sending files between members of the community, we will use GitHub to upload and distribute our code. This will allow us to see how the code is progressing and collaborate on it. (GitHub Guides, Hello World, 2020). GitHub will be beneficial to our group while we develop our Web Application as it will allow us to share the code amongst the group. It will be the Hub for us to upload, edit and document all coding related activities. GitHub is perfect for this as it provides the feature to allow all users to look at and edit the code, and to see any preview edits that have been made to the code. Hence, GitHub is required for us to finish our group project. https://github.com/danielmelfa-rmit/boomerang

SQL Server	Microsoft's SQL Server client server. The Microsoft relational database management system (RDBMS) is a piece of software that mainly stores and retrieves data for other programmes. These programmes can run on the same machine or on a different one.
Django (Version 3.1.7)	Django is the Web Framework that we will be using to build the front and back end of our Web Application. Django is a widely used framework that is free and open source. Django uses Python as its coding language, and it boasts a swift, efficient, and practical design. The reason we decided to use Django as it is serious about making ideas take fruition in a quick manner, it has good security, and the scalability of the framework is very quick and flexible. (Meet Django, n.d.) Django will be helpful to our group as it will provide the front-end and back-end framework for our Web application. This framework will allow us to code the website more efficiently and more effectively allowing us to produce a great end product. Thus, Django is required for us to finish our Web Application.
Python 3.9.0	We will be using Python to code the front and back end of our Web Application. Python is a free and open-source coding language that is a popular language to code with. This is due to it being a high-level programming language that is interpreted, object-oriented with dynamic semantics. Python is simplistic and easy to learn, that highlights on user readability which lowers the need for program maintenance. Due to it being so popular it also has a wide range of resources that our group can take advantage of when learning and creating a Web Application. Our team will be using Python as it is the language needed to code in collaboration with Django. (What is Python? n.d.) Python will be advantageous to our group project as it gives us an easy and efficient language to code the Web Application in. Django, our front-end and back-end framework requires the use of the Python language to function. Therefore, Python is required for us to finish our main project.
Atom	Atom is the source code editor that we will be using to code our Web Application. Atom is an extensively used source code editor that is free and open source. With the installation of a few plugins, we can use Django in Atom with Python as our coding language. Atom features GitHub integration so that it will allow us to push code to our GitHub Repository with ease.

	Atom will be beneficial to our group as it will provide us with a source code editor to code the Web Application in. It will be the editor that we will install Django into and use the programming language Python. Atom is ideal for this as it also has easy to use GitHub integration so we can upload to a repository with ease. As a result, Atom is required for us to finish our Web Application.
Deployment Tools	
Django	Django is a robust web platform for deploying Python apps and websites. Authentication, a custom database Object-Relational Mapper, and an extensible plugin architecture are all included in Django.

4. Challenges and Learning

Daniel Melfa

The “Building IT Systems” subject is the fourth subject that I am studying at RMIT, all the previous subject/courses that I have studied to date had its challenges, most of them being of the technical nature. This course although has been by far one of, if not the most challenging to date.
When people think of typical challenges of an average Information Technology project it is safe to assume that most people think of the multitude of technological challenges that can trouble any given project. Typical challenges may include poor software documentation, poor solution design documentation, ambiguous system features, and the list goes on.

While I have been working in the IT industry for a while, most projects I have worked in the past were with people that I had already known for a while, the requirements and outcomes were clearly decided from the beginning (from the time the solution was sold to a client), these factors made solution implementations somewhat easier.

The challenges that I found particularly difficult in this course were in the beginning related to getting 5 team members that did not know each other previously, people from different professional and personal backgrounds to agree on the initial decisions that had to be made early in the course.

Our group was a late formed one, the group as in its current form was only finalised late on week 2, towards beginning of week 3. In the first 4 weeks of the course, we did not have a team leader elected and all “negotiations” were somewhat confusing, it took us a while to decide what project stream we would follow; once the project stream was agreed on, it again took us another while to decide on what exactly we would do, how things would need to be done, etc. It felt like to me we were slowly falling behind schedule right on the most basic stages of the project.

As the weeks passed by, we established a working culture where everybody was encouraged to speak no matter what, everyone’s suggestions were debated within the group, and all decisions were made only after considerable deliberation.
This culture had a significantly positive impact in the team, it feels like to me everybody felt “at east” during team meetings and every team member was eager to contribute their best.
My learning out of this challenge is that paying attention to the human dynamics of any project is of utmost importance, in my opinion it is as important as the technical decisions themselves.
Aside from the “human challenges” of the project I now realise that some people believe that to work in an “Agile mode” is easy and that it requires no governance, but I learnt that this statement could not be further from the truth. To work in an Agile mode, you are required to constantly assess how things are progressing, you need to have a good

situational awareness of all aspects of the project, what parts are working well, and what parts are not working well.

Specifically related to the work that our group has done to date, while I believe that we followed the skills we learnt in the micro-credentials (and all course supporting materials) to the best of our abilities, looking back I ponder whether we spent too much time on the certain aspects of the project (i.e., technical documentation) and too little time on the “hands-on” side of things, learning the technology required for the project stream we chose and having a more significant attempt at coding our web application.

The learning for me is that it takes practice to be proficient in working in an agile team, and that working agile requires constant and continuous project re-assessment, and team dynamics re-assessment.

Additional learnings that I enjoyed in this course, seeing firsthand how important a Kanban style project board is (i.e., the Trello platform we used), how important a good team culture is, and I really found the micro credentials to be useful for my real-life job.

Megha Patel

Group Challenges:
The most difficult aspect was learning new tools and technologies in such a short amount of time. Most of us had no prior experience with the tools used to create Boomerang. I discovered that as tools and technology evolve, information can quickly become out of date, as I discovered with the tutorial I was following, where the code was written for a previous version and did not function on the most recent version. As a result, it is not only necessary to learn new tools and technology, but also to stay current with changes.

The group faced a few other challenges, such as changing the tool (React) that was determined in the first assignment. We decided to use Django for the project after some research. Due to the time constraints, it was difficult to learn and understand tools and language and implement the learning to the project. Our project necessitated a deeper understanding of each platform, which we attempted to obtain, but we encountered some difficulties, such as creating a database using SQL server, but then we discovered Django has its own SQLite.
Working remotely when everyone's schedules were all different was also a challenge.

How the Challenges were Addressed:
One thing we did as a group was pair up with someone who had some experience and was inquisitive about the areas. The best way to approach the challenge was to learn the tools and apply them to our project. The group decided to work on individual MVFs of their choosing because everyone wanted to learn, code and ‘wanted to get their hand dirty’. This way, the group will share the responsibility while also assisting in the advancement of the project.

We were unable to integrate the work – SQL server to Django, but we'll keep looking until we find a new way to stack the work.

Learning for the challenges:

The most important thing I learned was how a well-functioning successful group works and how their expertise and experience support each other. Despite not developing the successful MVF of search, I was able to get Django and Python to work.

I learned that not all tutorials and designs are straightforward, and to allow for more research and take it one step at a time when there are no crucial deadlines. I was effective in creating a 'Hello Page.' I gained a better understanding of database architecture and was introduced to GitHub, Python, and Django and few other tools and technologies. Project management and business analysist are equally important roles as designers, and I believe I performed admirably in those roles.

As a group we learnt Agile Ways of Working which is to bring people, processes, connectivity and technology, time, and place together to find the most appropriate and effective way of working to carry out a particular task. It is working within guidelines of the task but without boundaries of how we achieve it.
we were expose to new environment – working remotely, new tool and technologies and new people.

Changes:

Over the course of the project the team decided not to change much from the initial plan. The two main thing that changed were using only Python for the platform programming, the Python languages was more suitable and easier to learn for the chosen platform. We decided to use SQL but after coming up to halfway we realised Django has its inbuilt SQLite.
Project Plan Refinements

If the group were to do another project, I would put more time in to research the tool and technologies. The main challenges with our project were choosing the platform and programming languages and learning how to work with the software as we didn't have any previous experience. I believe that if the group had more time, we could have completed the website.

Timeline Refinements:

Considering the challenge of never having used a Django, I would allocate more time to the initial research and planning stages of the project. The major change I would have made to the timeline would be to start developing the MVFs in early stages. This would allow extra time to source any changes or items required and pick up any issues earlier, like the SQL and SQLite. This would also give time for the development of an MVF.
Risks and Unexpected Events

The main risk we faced was the developing side of the project, as we were all new to the using Django and programming. Everyone had to start from the beginning, so building the MVFs took up the significant amount of time. This was especially apparent when we tried to connect the SQL database to Django which is also through our lack of experience with the platform. We did mitigate risks by having our strongest team members on work on the features and prototype and working in pairs.

Tyson Carroll

Group challenges:
I believe initially as a group we had an issue with spitting up tasks between members of the group, creating an appropriate timeframe in correlation to our current skill sets and how much we needed to learn. Working as a team remotely is also another challenge, we had to face during this project which can make it harder for the progression of the project. These challenges were expected when we first started this project as it was the first time we actually worked on a Web application, met each other and worked together. Also, the remote learning aspect of the course cannot change so it was a challenge we had to deal with appropriately.

Over time of the project, we discovered that some of the tools we initial thought would be needed for the project were not actually required to get the stuff done that we did. The main tool that we culled was React. We planned on using React for the front-end side of our web application, however, we decided to use Django for our front and back end while adding HTML into the mix as well. Also, with how substantial the tools we used were like Python and Django, there was just so much to learn in the given timeframe that we had. Therefore, we did not obtain the full girth of the knowledge possible if more time was given. Also, another issue we had was we created two different databases one in SQL and the Django default database was SQLite.

How the challenges were addressed:
We were able to fix the issue of splitting up tasks over time as we were able to play to each other’s strengths are we knew each other more. We were able to divide tasks to people who were stronger in certain types of areas so that we could push out a good project. Due how substantial the tools we used were, we decided to focus a lot on the learning aspect and just having a crack at things we were able to code with what we learnt. We ensured that the timeline to get tasks done was progressively more realistic with the speed we obtained knowledge and understood how to complete aspects through the continuation of the course.
The main feature that is still being developed is moving the SQL database into the SQLite database and piecing the other aspects like the homepage into the Django Framework so that it can all work together.

Learning from the challenges:
The group learnt a lot through the challenges we faced through building our Web Application. We learnt better time management skills, being able to set appropriate times to complete tasks and make sure they are done prior to the due date. We learnt

basic Teamwork skills being able to work as a team and work together to get tasks done. Which these helped us learn how to plan a project more effectively for the future. The group also learnt the basics of Python and HTML coding, fundamentals of the Django framework, also how SQL and SQLite Databases work and are used.

Changes:
Production of the project took longer than we expected as we needed to put more time into learning how to use the tools that we chose to develop our Web Application. Therefore, we allocated more time to tasks that we originally thought would be less time consuming than we expected. The timeline of the full project completion would be past the given timeframe allocated to us for this course.

Project plan refinements:
Now that I have experienced the development side of things in terms of learning Python, HTML, and the Django framework. I believe the team should have reduced the difficulty of the Minimum Viable Features (MVF) to features that were more of our skill level at the time. Due to us not having prior experience I believe it was hard to determine what MVFs the group would be able to complete at the start of the project. For example, a MVF we could have had instead is to create a registration process for users, which was completed.
Timeline refinements: Describe any changes you would make to your project timetable and reasons why.
If the team had prior extensive experience using python, HTML, Django and SQL/SQLite databases I believe only a minor extension, or no extension of the given time would be needed as it would not take as long to develop a Web application. However, due to the lack of experience of the team as a unison in correlation to the tools we used and working in a properly managed team. I believe that if we had more time to learn the tools previous to doing the project, we would be able to complete more aspects of the Web Application. Therefore, an extension of the timetable would have helped us complete more of the project.

Moustafa Al-Meahi

The most troublesome perspective was learning new instruments and advances in a short measure of time. The majority of us had no related knowledge with the instruments used to make Boomerang. I found that as devices and innovation advance, data can immediately get obsolete, as I found with the instructional exercise I was following, where the code was composed for a past adaptation and didn't work on the latest form. Therefore, it isn't simply important to learn new instruments and innovation, yet additionally to remain current with changes.

We had the option to fix the issue of separating errands over the long haul as we had the option to play to one another's qualities are we knew each other more. We had the option to separate assignments to individuals who were more grounded in specific sorts of territories so we could push out a decent undertaking. Due how generous the apparatuses we utilized were, we settled to zero in a ton on the learning viewpoint and simply having a figure out at things we had the option to code with what we realized. We guaranteed that the timetable to complete assignments was continuously more sensible with the speed we acquired information and saw how to finish perspectives through the continuation of the course.

The principle highlight that is as yet being created is moving the SQL information base into the SQLite data set and piecing different viewpoints like the landing page into the Django Framework so it would all be able to cooperate.

Explicitly identified with the work that our gathering has done to date, while I accept that we followed the abilities we mastered in the miniature qualifications (and all course supporting materials) as well as could be expected, thinking back I contemplate whether we invested a lot of energy in the specific parts of the venture (i.e., specialized documentation) and too brief period on the "active" side of things, learning the innovation needed for the task stream we picked and having a more critical effort to code our web application.

The learning for me is that it takes practice to be capable in working in a spry group, and that working coordinated requires steady and persistent undertaking re-evaluation, and group elements re-appraisal.

Extra learnings that I delighted in this course, seeing firsthand how significant a Kanban style project board is (i.e., the Trello stage we utilized), how significant a decent group culture is, and I truly discovered the miniature qualifications to be helpful for my genuine work.

Patrick Jenner

Group challenges

At the start we had a few competing ideas until we narrowed it down to one. The was a warehouse planning tool I called Speci-fi. It would have been designed to help companies plan out a warehouse and included things they hadn’t before considered. The other idea was a discord bot. The idea of Boomerang was pitched by Daniel, which was rental platform.

At first Django was chosen to do the website. This being the first modern website most of us had worked on, we didn’t immediately realise that it required both a front and back end. Not having done much HTML I felt I had to learn quite quickly. I discovered that to make the front end of the website look contemporary I need to learn CSS.

How the challenges were addressed:
We decided on Daniel's idea because it was the one that had the most enthusiasm and uniqueness behind it. After much research and discussion with the team, it was decided that python would be used for our backend and HTML for our Front end. I had done some HTML before, but my knowledge was a little out of date and thanks to the subjects 'Into to programming HTML was I was able to do some HTML tutorials on https://www.w3schools.com and discovered that CSS was required to build a website with a modern aesthetic.

Learning from the challenges:
We now have a better idea of how to create a website which relies on SQL database and external backend software. I have acquired new skills such as CSS, HTML, Python and SQL. More importantly I've learnt how to be a team player and how to work effectively on a team project.

Changes:
Not have created or published a website before I assumed that only python could be used to build a website, but we need other languages including HTML and CSS.

Project plan refinements:
More HTML and CSS learning from an earlier point in time. Our Figma prototype was the basis for all functional and aesthetic aspects. However, due to my inexperience, the time needed to replicate it in CSS, would have taken much longer than expected.

Timeline refinements:
We would include more room for unexpected risks and occurrences due to our overall lack of experience. Importantly we should have better accounted for our lack of experience and the time it would take to learn the skills. We would have hired experienced individuals who would have been much quicker and had more clear direction to move in.

5. Marketing Pitch

In 2017 a study concluded by the Office of Local Government, displayed that roughly 74% of residents in New South Wales have machinery and other household Items laying around the house and not in use. These people who have machinery laying around are the ones who we seek to attract for our platform. Not only is their machinery taking up space, but it also brings them no benefit. We attempt to change that by providing a platform for them to list their goods on for lease.

Our Product ‘Boomerang’ will solve this issue by providing a platform for owners and businesses to list their machinery and any other good, online for hire. This will benefit businesses and owners in multiple ways, by giving them more space around the house and business. It will also earn them some extra money for the items that are sitting around. Success for our platform will come from the competition that will exist within our platform. Owners competing with each other on leasing their goods off.

6. Skills and Jobs

In order to run our platform after a successful investment we require candidates to fill the following roles:

- Front end Web developer
- Full stack programmer/developer
- Project Manager
- Marketing/Sales manager

Front end Web developer

We'd ideally like an innovative thinker who can think outside the box and has a passion for their work. We're looking for someone who shares our values and our vision for 'Boomerang' and can take direction and feedback. Ideally, they will have a positive attitude, attention to detail which will be key in this role.

Our frontend developer will be someone who has enthusiasm for the platform as well as enough experience to make our site look as close to our design prototype as possible while being as compatible as possible. They will be family with versioning platforms such as git hub. They will have confidence and have a strong feel for how to plan the required web platform. We want our platform to run as efficiently as possible and have a quick and responsive UI.

They will have:

- Solid knowledge of HTML and CSS
- Experience with web development for both desktop and mobile platforms
- Experience working in a team environment
- Experience with Python is a plus
- Knowledge of user centred design
- Clear communication skills
- Understanding of Figma and/or Adobe XD
- Will have 2 years experience
- Experience with Django is a plus, but not required
- Able to work with web form integration

The job is full time, 9-5, 5 days a week. Will offer a competitive salary and future opportunities. You will have the opportunity to build the company with us and become our head of web development in the future.
(101 Web Technology, 2021)
(Beaumont People, 2021)
(Net Search, 2021)

Full stack programmer/developer

We are looking for someone to join our team who understands and shares our goals. Our platform is rapidly expanding which means we need someone who can adapt quickly to the changing environment and can work proactively while making suggestions and creating solutions for any technical issues that may occur.
Our programmer will be responsible for the overall scope of platform. They will help us expand our Django framework and platform. They will ideally have experience in Python and SQLite. They will be technically minded and have excellent verbal and written communication skills. They will need to liaise with our server hosting provider and make sure our platform is up and operational as well as responsible for making sure the platform is quick and responsive at all times. Programmer/ Developer needs to be someone who knows how to use Django and python.

- Fluent in SQL
- Knowledge of HTML and CSS is a plus
- Experience with web development for both desktop and mobile platforms
- Experience working in a team environment
- Experience with Python is required
- Knowledge of user centred design is plus
- Clear communication skills
- Will have 2 years experience
- Experience with Django is required
- Able to work with web form integration

The position is full time, 9-5, 5 days a week. Will offer a competitive salary. This position will offer the opportunity for future leadership opportunities to lead the team as the company expands.

(Bendigo Bank, 2021)
(Department of Industry, 2021)
(Solutions Connect Pty Ltd, 2021)

Project Manager

Our platform ‘Boomerang’ is unique and will be scaled with rapid growth. This means we will require someone who can adapt to the changing nature of the processes and develop new procedures, while motivating and communicating with the marketing and technical teams.

Our platform is complex, and a large amount of complicated co-ordination and organisation is required to keep everything and everyone on the right track towards our goals. Our project Manager will need to discuss and co-ordinated with our team members and make a note of what work needs to be done and make decision on how to accomplish certain goals. They will be dynamic and able to solve problems with little assistance.

We require our manger to understand the entire scope of the project and the goals involved. They will ideally have strong interpersonal skills and very strong communicational skills. They will also require a strong business background and experience with building businesses.

- Experience with business development
- Demonstrate experience goal orientated projects
- Experience developing procedures and communicating with teams
- Clear communication skills
- Will have 5 years’ experience
- Knowledge SQL is a plus
- Knowledge of HTML and CSS is a plus
- Understanding web development for both desktop and mobile platforms
- Experience working in a team environment
- Experience with Python is required
- Knowledge of user centred design is plus
- Clear communication skills
- Will have 2 years’ experience
- Experience with Django is required

The role is full time, 9-5, 5 days a week. This is an opportunity for someone to build the business and put their mark on a new and exciting platform with a lot of potential. We offer a competitive salary and the freedom to work your way, expand your responsibility and gain more experience.

(ACT Digital, 2021)
(Project Partners, 2021)
(Public Sector People, 2021)

Marketing and Sales Manager

We need a Marketing Manager who is creative and shares our passion and vision for the ‘Boomerang’ platform. This role is key to the success of our platform. We require someone who can operate in a business to consumer capacity.
You will be able to analyse the market and have a strong understanding of the treads on social media to attract customers to our platform. You will have the ability to know how to anticipate and address the concerns of any potential customers. You will be able to plant the seed of our service in their minds and demonstrate the need for our platform. Your communication skills will be paramount in the role as we will require you to develop marketing strategies and campaigns. We need someone who understands our platform so they can reach our target market by inventing creative ways to reach them.

- Advances written communication skills
- Experience with business development
- Demonstrate experience goal orientated projects
- Proven Management of SEO
- Previously successful Marketing
- Experience working with teams
- Proven sales record
- 5 years experience
- Social media experience is a must

We offer a retainer and commission and as well as future leadership opportunities regarding new sales team members and the opportunity advance your career. This opportunity is full time, 9-5, 5 days a week. This roles pays competitive salary.

(Club Helensvale, 2021)
(Audeara, 2021)

References

1. 101 Web Technology. (2021). Retrieved from Seek: https://www.seek.com.au/job/52218733?type=promoted#searchRequestToken=daf40050-3763-4241-8c3a-d0145f9f4313
2. ACT Digital. (2021). *Seek*. Retrieved from https://www.seek.com.au/job/52093869?type=promoted#searchRequestToken=4429d0ab-7335-4485-8e4d-c0c81fb8e9cf
3. Audeara. (2021). *Seek*. Retrieved from https://www.seek.com.au/job/52335093?type=standout#searchRequestToken=498f6644-e9ef-4855-8555-6c753457c7e6
4. Beaumont People. (2021). *Seek*. Retrieved from https://www.seek.com.au/job/52339430?type=standard#searchRequestToken=daf40050-3763-4241-8c3a-d0145f9f4313
5. Bendigo Bank. (2021). Retrieved from Seek: https://www.seek.com.au/job/52324847?type=standout#searchRequestToken=09650549-2c2d-42e1-8fd7-5f7cd80a22a1
6. Club Helensvale. (2021). *Seek*. Retrieved from https://www.seek.com.au/job/52144307?type=promoted#searchRequestToken=498f6644-e9ef-4855-8555-6c753457c7e6
7. Department of Industry. (2021). *Seek*. Retrieved from https://www.seek.com.au/job/52316316?type=standard#searchRequestToken=09650549-2c2d-42e1-8fd7-5f7cd80a22a1
8. Net Search. (2021). *Seek*. Retrieved from https://www.seek.com.au/job/52323710?type=standout#searchRequestToken=daf40050-3763-4241-8c3a-d0145f9f4313
9. Project Partners. (2021). *Seek*. Retrieved from https://www.seek.com.au/job/52279119?type=promoted#searchRequestToken=4429d0ab-7335-4485-8e4d-c0c81fb8e9cf
10. Public Sector People. (2021). *Seek*. Retrieved from https://www.seek.com.au/job/52346669?type=standard#searchRequestToken=4429d0ab-7335-4485-8e4d-c0c81fb8e9cf
11. Solutions Connect Pty Ltd. (2021). Retrieved from Seek: https://www.seek.com.au/job/52077499?type=standout#searchRequestToken=09650549-2c2d-42e1-8fd7-5f7cd80a22a1