



---

## UniLodge

An AirBnB Alternative for Students

---

### Final Year Project B.Sc.(Hons) in Software Development

BY  
FARIS NASSIF  
AARON BURNS

APRIL 28, 2020

Advised by **Dr. John French & Dr. Martin Kenirons**  
DEPARTMENT OF COMPUTER SCIENCE AND APPLIED PHYSICS  
GALWAY-MAYO INSTITUTE OF TECHNOLOGY (GMIT)

# Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Introduction</b>                              | <b>3</b> |
| 1.1      | Objectives of the Project . . . . .              | 3        |
| 1.2      | Summary . . . . .                                | 4        |
| 1.2.1    | Methodology . . . . .                            | 4        |
| 1.2.2    | Technology Review . . . . .                      | 4        |
| 1.2.3    | System Design . . . . .                          | 4        |
| 1.2.4    | System Evaluation . . . . .                      | 4        |
| 1.2.5    | Conclusion . . . . .                             | 4        |
| <b>2</b> | <b>Methodology</b>                               | <b>6</b> |
| 2.1      | Preliminary Research and Project Onset . . . . . | 6        |
| 2.1.1    | Initial Meeting and Brainstorming . . . . .      | 6        |
| 2.1.2    | Methodology Consideration . . . . .              | 7        |
| 2.1.3    | Determining the Methodology . . . . .            | 7        |
| 2.2      | Version Control . . . . .                        | 7        |
| 2.2.1    | Considerations . . . . .                         | 7        |
| 2.2.2    | Github . . . . .                                 | 7        |
| 2.3      | Testing . . . . .                                | 8        |
| 2.3.1    | Types of Testing we'll do . . . . .              | 8        |
| 2.4      | System Architecture . . . . .                    | 8        |
| 2.4.1    | Components of Project . . . . .                  | 8        |
| 2.4.2    | Brief discussion about moving pieces . . . . .   | 8        |
| <b>3</b> | <b>Technology Review</b>                         | <b>9</b> |
| 3.1      | Initial Considerations . . . . .                 | 9        |
| 3.1.1    | MEAN Stack . . . . .                             | 9        |
| 3.1.2    | VueJS . . . . .                                  | 11       |
| 3.1.3    | Redis . . . . .                                  | 11       |
| 3.2      | Chosen Technologies . . . . .                    | 11       |
| 3.2.1    | Angular . . . . .                                | 12       |
| 3.2.2    | The Flask Micro-framework . . . . .              | 12       |
| 3.2.3    | MongoDB . . . . .                                | 12       |
| 3.2.4    | ExpressJS / Node . . . . .                       | 12       |
| 3.3      | Deployment . . . . .                             | 12       |

|          |                                      |           |
|----------|--------------------------------------|-----------|
| 3.3.1    | Heroku                               | 12        |
| 3.3.2    | Honcho                               | 12        |
| 3.3.3    | Ngrok                                | 12        |
| <b>4</b> | <b>System Design</b>                 | <b>13</b> |
| 4.1      | Overview                             | 13        |
| 4.2      | Data Tier                            | 13        |
| 4.2.1    | MongoDB                              | 13        |
| 4.2.2    | Schema                               | 13        |
| 4.2.3    | Hashing                              | 13        |
| 4.2.4    | Another thing i'm forgetting         | 13        |
| 4.3      | Logic Tier                           | 13        |
| 4.3.1    | Login/Authentication                 | 13        |
| 4.3.2    | Web Token                            | 13        |
| 4.3.3    | Flask API                            | 13        |
| 4.4      | Application Tier                     | 13        |
| 4.4.1    | Angular Stuffs                       | 13        |
| 4.4.2    | Routing?                             | 13        |
| 4.4.3    | Something else ..                    | 13        |
| <b>5</b> | <b>System Evaluation</b>             | <b>14</b> |
| 5.1      | Testing                              | 14        |
| 5.1.1    | Types of Testing                     | 14        |
| 5.1.2    | Deployment Testing                   | 14        |
| 5.1.3    | Unit Testing?                        | 14        |
| 5.1.4    | System Testing                       | 14        |
| 5.1.5    | Some more things here I'm neglecting | 14        |
| 5.2      | Overall Evaluation                   | 14        |
| <b>6</b> | <b>Conclusion</b>                    | <b>15</b> |
| 6.1      | Overview                             | 15        |
| 6.2      | Learning Outcomes                    | 15        |
| 6.3      | Final Thoughts                       | 15        |
|          | <b>Appendices</b>                    | <b>17</b> |

# List of Figures

|     |                         |   |
|-----|-------------------------|---|
| 1.1 | UniLodge Logo . . . . . | 5 |
|-----|-------------------------|---|

# About this Project

**Abstract** Housing, and the lack of affordable accommodation has become a hot topic in recent times, especially in relation to students having to endure undesirable living conditions for even more so undesirable rates. Daily, students are commuting great distances to avoid having to endure the financial burden of living at a local level. Students being unable to bear this burden leads to lower admission and attendance rates, an undesirable outcome for both the educational institutions and those looking to attend.

The proposed solution to help bridge this issue will be a web application, providing an easily accessible platform for students, and for those living locally or living at a reasonable distance, who may not have the outlet to advertise their spare room or inhabited apartment.

**Authors** This was developed as a 15 credit project by Faris Nassif and Aaron Burns, final year students of Galway-Mayo Institute of Technology.

**Acknowledgements** The authors would like to acknowledge the project supervisors Dr. John French and Dr. Martin Kenirons for the time and advice they dispensed during the course of the project.

# Chapter 1

## Introduction

During the decision making process it was decided that the project must be relevant not only to the team but also to peers. The project must also hone existing skills and allow for the natural development of new techniques and processes while also being worthy in scope.

UniLodge was the result of much deliberation. UniLodge would serve students and home owners in Galway, allowing for both a practical and streamlined avenue of accommodation advertisement while existing as a simple to use platform for identifying listings that fit the standards and requirements of the student.

### 1.1 Objectives of the Project

As previously mentioned, the main objective of the project is to create an application that would help bridge the gap between tenants and students by providing both parties with a platform that would allow for the organization of accommodative housing services specifically for students in the Galway area. In order to reach that goal, by extension, other objectives had to be set out to enable TODO MIGHT GET RID OF THIS

- Evaluate and investigate the frameworks and tools available for creating a platform independent web application.
- Create and develop an application that will allow users to arrange or offer lodging services for students.
- Identify and compare applications of a similar nature, critically analyze those alternatives and apply any beneficial findings to our project.
- The application will, at a minimum, allow users to register an account, login, post listings and communicate with other users via a commenting system.

## **1.2 Summary**

This section will contain a brief overview of each chapter outlined in this dissertation.

### **1.2.1 Methodology**

In this chapter, the processes undertaken during the life cycle of the project in regards to planning and development will be outlined. The decisions, thought processes and influential factors leading up to those processes and design implementations will also be described.

### **1.2.2 Technology Review**

A technological review will attempt to encapsulate the technical aspect of the project. This includes the different technologies incorporated, their implementation, why they were implemented and why they were chosen. The benefits of the chosen implementations will be critically analysed and compared with alternatives.

### **1.2.3 System Design**

A detailed explanation of the overall architecture of the project will be provided. Code-snippets and diagrams will be included to help illustrate the inner workings of the application at a high level. Improvements to the system will be identified and potential competitive alternatives will be discussed.

### **1.2.4 System Evaluation**

An evaluation of the software developed in the project will be carried out with the initial project objectives in mind. The final results of the project will be reviewed, including an analysis of areas for improvement and potential changes applicable to the overall system.

### **1.2.5 Conclusion**

To conclude, a brief review will encapsulate the overall system. Key insights will be identified and reflected upon. A final analysis will describe the overall experience and what was learned from the development life-cycle of the project.

Figure 1.1: UniLodge Logo





## Chapter 2

# Methodology

### 2.1 Preliminary Research and Project Onset

The first project meeting began in the final week of September, briefly after the project requirements had been assigned and outlined. An early start was agreed to be something that would greatly benefit the overall development of the project and it was concluded that an idea should be finalized as soon as possible to allow for necessary pre-development research.

This section will further explore the aforementioned pre-development process, the influence of the supervisory meetings on this stage, the overall methodical conclusion and its influence on the initial project architecture.

#### 2.1.1 Initial Meeting and Brainstorming

In the weeks before development began, after the project idea had been finalized, technologies, concepts and potential inclusions were explored and discussed between the team members. A brainstorming phase was conducted on what to incorporate into the project.

##### Brainstorming

The team members met prior to the initial supervisory meeting and discuss potential avenues of exploration during the development phase. To produce effective ideas, questions had to be asked relating to the ultimate goals and objectives of the project, these included:

- *What research areas should be prioritized before the development phase is initialized?*
- *What type of Methodology would best fit our approach?*

- *What benefits would different methodologies have when compared to others?*

### 2.1.2 Methodology Consideration

There were numerous possible methodologies to consider, namely Waterfall, Rapid Application Development and Agile to name a few. Following discussions internally between the team members and talks with supervisors, the list of potential methodologies were shortlisted to both **Waterfall** and **Agile**.

### 2.1.3 Determining the Methodology

Following the decision to shortlist both Waterfall and Agile, research began on which route would be best to take in relation to the scope and overall goals of the project.

## 2.2 Version Control

Blah..... was cited by [3] in ... You should refer to images and tables by their label and let latex figure out the numbering for you. E.g. we can refer to the figure on this page as Fig.1.1 instead of writing "Fig.1"...Blah..... was cited by [3] in ... You should refer to images and tables by their label and let latex figure out the numbering for you. E.g. we can refer to the figure on this page as Fig.1.1 instead of writing "Fig.1"...

### 2.2.1 Considerations

Blah..... was cited by [3] in ... You should refer to images and tables by their label and let latex figure out the numbering for you. E.g. we can refer to the figure on this page as Fig.1.1 instead of writing "Fig.1"...

### 2.2.2 Github

Github was used as the chosen method of version control for the project. A Git repository was setup remotely and used during development to allow for collaboration, code security and to track the progress of the project as well as providing the functionality of managing dependencies and providing alerts if new versions should arise.

## **2.3 Testing**

### **2.3.1 Types of Testing we'll do**

## **2.4 System Architecture**

### **2.4.1 Components of Project**

### **2.4.2 Brief discussion about moving pieces**

## Chapter 3

# Technology Review

Over the course of the project life cycle a plethora of frameworks, tools and development applications were available for integration or use with our application. This section aims to discuss the tools and technologies that were heavily considered and those that were ultimately used, why they were chosen and what alternatives were available.

### 3.1 Initial Considerations

During the discussion and planning phase goals were outlined and proposed however, how to reach the end point was still very ambiguous. For this reason a lot of time was spent considering different approaches and uncovering the benefits and drawbacks of venturing down a chosen route. This brief section will outline those initially considered approaches.

#### 3.1.1 MEAN Stack

The MEAN Stack combines the best of JavaScript based technologies. The Stack is essentially a collection of open source components that provide a streamlined environment for building dynamic web applications.

The MEAN Stack consists of:

- MongoDB
- ExpressJS
- Angular
- Node.js

Perhaps the greatest attribute of the MEAN Stack for developers is that it's essentially a single language development stack, which can also be one of its most undesirable attributes depending on the developers JavaScript competency [3]. Other attributes that the development team considered attractive were the vast array of libraries and modules exposed via Node, its speed, usability and flexible structure.

Another technology stack that piqued the attention of the developers was the MERN Stack, which is essentially the MEAN Stack excluding Angular and including React. Research was conducted on comparing the two [4] and the following was found:

| Angular  | React  |
|--|--|
| <ul style="list-style-type: none"> <li>+ Testing tools like Jasmine and Karma are well documented Angular frameworks that allow for seamless human-readable Unit Tests or browser/platform based test cases.</li> <li>+ Application logic is a lot clearer and less convoluted than React due to its declarative nature.</li> <li>+ Enforces MVC-like design, giving developers an underlying structure to adhere to. React applications can be harder to maintain considering the overall design can be ambiguous and more unstructured.</li> <li>+ Unidirectional data flow in applications allow data to flow to more seamlessly check for a change of state.</li> <li>- Weak ability to debug code. Debugging can be ambiguous without manual inclusion of libraries.</li> </ul> | <ul style="list-style-type: none"> <li>+ Mastering React is a lot less punishing than delving into Angular, Angular being a complete framework that incorporates associated knowledge of concepts like MVC or familiarity with Typescript.</li> <li>+ Unidirectional data flow in applications allow data to flow to more seamlessly check for a change of state.</li> <li>+ Very lightweight and less cumbersome than Angular for setup and collaboration. Dependency control is managed automatically.</li> <li>- Relies heavily on third-party libraries for actions and tasks that Angular could perform by on the fly due to its built in service wrappers like for example Angular's built in wrappers for HTTP calls to the backend.</li> </ul> |

Table 3.1: Advantages and Disadvantages of React & Angular

### 3.1.2 VueJS

VueJS is a JavaScript based framework used for building user interfaces and single page applications that can integrate seamlessly into a project at any stage. VueJS is marketed as an approachable, versatile and performant framework, boasting an incrementally adoptable system that's scaleable between a fully featured framework and a library [2]. However a major downfall of the framework is considered to be its steep learning curve. VueJS was trialed for two weeks by the team, and following supervisory meetings and discussions between team members, the team ultimately decided due to the steep learning curve, the intimidating documentation for beginners and lack of tutorials that it wouldn't see a place in the development stack.

### 3.1.3 Redis

Redis, meaning REmote DIctionary Server is an in-memory distributed key-value data-store. Redis supports multiple types of data structures including, streams, bitmaps, sets and spatial indexes to name a few [5]. Key value databases excel at providing rapid access to information that has a corresponding function and following research and discussions with supervisors Redis was initially a very attractive inclusion into the project. The use of in-memory storage offers a number of advantages, namely data retrieval which as mentioned previously is extremely fast as well as memory writing being performed in mono-thread allows the write to be isolated, avoiding data loss [1].

At early stages in the project, various machine learning implementations were being considered. Redis would have been a perfect inclusion should the project have adopted artificial intelligence in any form, allowing for rapid access and storage of short-lived large scale machine learning data.

## 3.2 Chosen Technologies

Following research, input from supervisors and trials of the aforementioned technologies, a stack was constructed that the developers felt would fit both the objective and scope of the project. This section will outline the technologies, tools, languages, frameworks and concepts that were ultimately implemented and descriptions of relevant implementations will be illustrated at a conceptual level.

- 3.2.1** Angular
- 3.2.2** The Flask Micro-framework
- 3.2.3** MongoDB
- 3.2.4** ExpressJS / Node
- 3.3** Deployment
  - 3.3.1** Heroku
  - 3.3.2** Honcho
  - 3.3.3** Ngrok

## Chapter 4

# System Design

### 4.1 Overview

### 4.2 Data Tier

#### 4.2.1 MongoDB

#### 4.2.2 Schema

#### 4.2.3 Hashing

#### 4.2.4 Another thing i'm forgetting

### 4.3 Logic Tier

#### 4.3.1 Login/Authentication

#### 4.3.2 Web Token

#### 4.3.3 Flask API

### 4.4 Application Tier

#### 4.4.1 Angular Stuffs

#### 4.4.2 Routing?

#### 4.4.3 Something else ..



## Chapter 5

# System Evaluation

### 5.1 Testing

Approaches to testing something something

#### 5.1.1 Types of Testing

#### 5.1.2 Deployment Testing

#### 5.1.3 Unit Testing?

#### 5.1.4 System Testing

#### 5.1.5 Some more things here I'm neglecting

### 5.2 Overall Evaluation

## Chapter 6

# Conclusion

### 6.1 Overview

### 6.2 Learning Outcomes

### 6.3 Final Thoughts

# Bibliography

- [1] Younes Kkhourdifi Alae El Alami, Mohamed Bahaj. Supply of a key value database redis in-memory by data from a relational database. *International Journal of Innovative Research in Computer and Communication Engineering*, pages 49–50, 2019.
- [2] Kostas Maniatis Alex Kyriakidis and Evan You. The majesty of vue.js. pages 6–7, 2016.
- [3] Dantala O. Oyerinde Bakwa D. Dunka, Edim A. Emmanuel. Simplifying web application development using - mean stack technologies. *International Journal of Latest Research in Engineering and Technology*, 04:70–74, 2018.
- [4] Pragati Bhardwaj. Analysis of stack technology: a case study of mean vs. mern stack. *International Journal of Innovative Research in Computer and Communication Engineering*, 06:3610–3614, 2018.
- [5] S Sanfilippo et al. Redis. URL: <https://redis.io/>.

# Appendices

## Source Code

<https://github.com/farisNassif/UniLodge>

## Heroku Web Application

<https://unilodge.herokuapp.com/home>

## Swagger API

<https://app.swaggerhub.com/apis/farisNassif/UniLodge/1>

## Screencast

TODO