Project Plan
Student Housing: Roomies

Lex de Kort

September 2022

Version History

Version	Date	Changes	State
1.0	September 19, 2022	Initial release	Incomplete
2.0	October 7, 2022	• Fill in details of sprint 3	Incomplete
3.0	November 25, 2022	Added CI/CD diagram	Incomplete

Distribution

Version	Date	Receivers
1.0	September 19, 2022	Frank Coenen & Marcel Boelaars
2.0	October 7, 2022	Frank Coenen & Marcel Boelaars
3.0	November 25, 2022	Frank Coenen & Marcel Boelaars

Contents

1 Introduction

1.1 Context

While studying at Fontys I have often heard many students complain about the difficulty of finding rooms and apartments in Eindhoven. There are very few websites, some of which are only available in Dutch, with some even listing rooms that have already been taken. And there are plenty of rooms and apartments that aren't listed on any websites, relying entirely on word of mouth alone. Other lodgings might have restrictions that you only hear about after you've gone through the application process, making the application a pretty big waste of time. These factors and more make for a very frustrating process that often leaves students defeated or just prevents them from studying here in the first place. For a city that prides itself in being a hub for international students for a variety of highly acclaimed universities, it is rather baffling that very little effort has been made to make this process a lot more friendly for both internationals and natives. As such, I have decided to create a web app that both internationals and natives can use to help find lodgings here in the beautiful city of Eindhoven.

1.2 Goal

As stated in the previous section, the purpose of the web app is to facilitate the process of helping students (both international and native) find lodgings here in Eindhoven. It will also allow landlords to display their offerings without having to rely on word-of-mouth or other means to get interested students.

1.3 Scope

Initially, the app will be limited to the city of Eindhoven. This will ensure the scope won't be too large, while also letting us be more flexible with feature testing and bug fixing. It will also reduce the system load, allowing us to figure out where optimization is necessary before a larger roll-out is possible.

2 Backlog

2.1 Terminology

In order to explain some of the terminology used in the user stories, I have made a small dictionary in which I explain what I mean with certain words:

Term	**Meaning**
Listing	An advertisement / post for a room
Response / respond	To sign up for a listing
Active	A listing that a student can respond to

2.2 User Stories

ID	User Story	Acceptance Criteria
US-01	As a student I want to be able to look at the available listings so that I can find a room that interests me	• Rooms are filterable based on surface area, neighborhood, price, etc. • Rooms are displayed based on user filters • Only available rooms are displayed
US-02	As a student I want to express interest in renting a room so that the landlord can contact me	• The landlord cannot contact the student without their contact info • The student should be informed when the listing is closed
US-03	As a landlord I want to be able to make new listings so that I can get tenants	• The landlord should be able to fill in all required room parameters (surface area, price, etc.)
US-04	As a landlord I can update a listing so that the information and details remain up-to-date	• All created fields can be modified

ID	User Story	Acceptance Criteria
US-05	As a landlord I want to be able to close listings so that I no longer get applicants for rented rooms	• The listing should be removed from the list view
US-06	As a user I can inspect my account details so that I can ensure that my information is up-to-date	• All created fields can be modified
US-07	As a landlord I want to be able to set a deadline on a listing so that I can limit the responses I might get	• There be a minimum and maximum limit on how long a listing can stay open
US-08	As a student I want to be able to communicate with the landlord so that I can contact him if I encounter any issues	• Communication should occur through the website
US-09	As a landlord I want to be able to easily select a student so that I can get into contact with them	• The landlord should be able to choose between several options on how the selection process is handled
US-10	As a student I want to be kept in the loop regarding new listings so that I don't have to check the website every day	• Only correspond with people that have signed up for the service • Correspondence should match an user's personal preferences
US-11	As an administrator I want to be able to check and approve new account registrations so that only people with valid credentials have access to certain features	• Validation of credentials should occur through an intuitive process

2.3 Prioritization

The priority of a user story will be given a number between 1 and 100, with 1 being low priority and 100 being high priority.
The story points determine how much time a story is expected to take to fully implement; the higher the points, the more time is expected for implementation.

Task	**Priority**	**Story Points**
US-01	90	9
US-02	85	7
US-03	65	7
US-04	25	6
US-05	25	2
US-06	25	2
US-07	40	3
US-08	50	6
US-09	45	5
US-10	40	8
US-11	20	1
US-12	70	8

3 Phasing

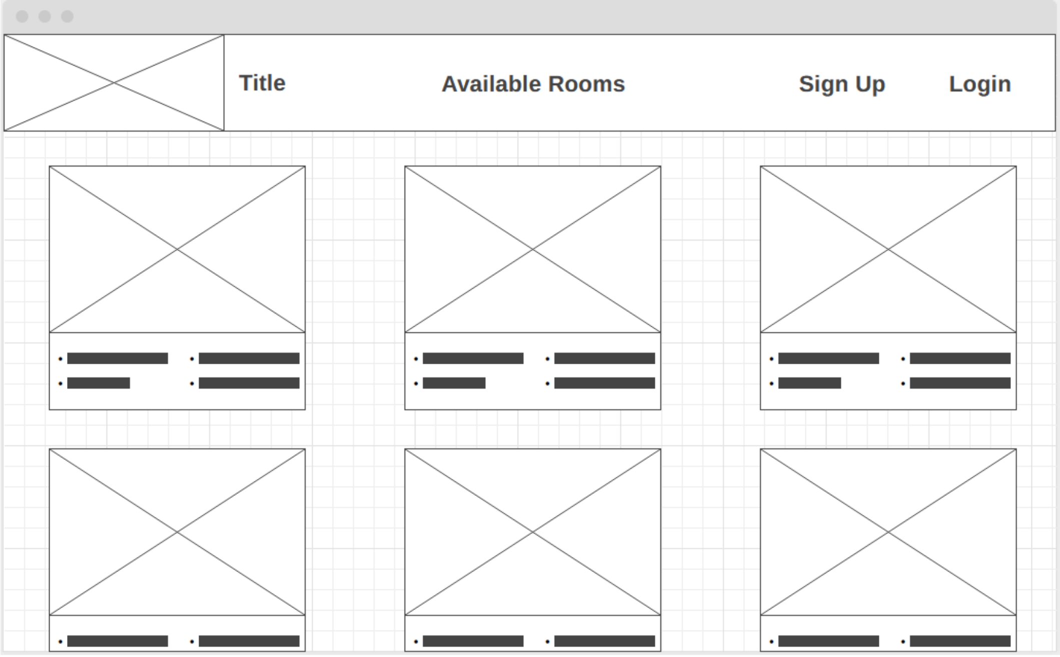
3.1 Sprint planning

Sprint 1	• Create project plan • Create URS • Create wireframes • Set up basic project • Start implementing US-01: 1. Create generic classs & interfaces necessary for retrieving listings 2. Create entity classes 3. Create mock listings for testing
Sprint 2	• Continue work on URS • Update wireframes • Start work on test plan • Finish implementing US-01 • Start implementing US-02: 1. Create student account classes 2. Create mock data for student accounts 3. Implement respond / react functionality • Write unit tests for US-01 & US-02 • Start building React front-end

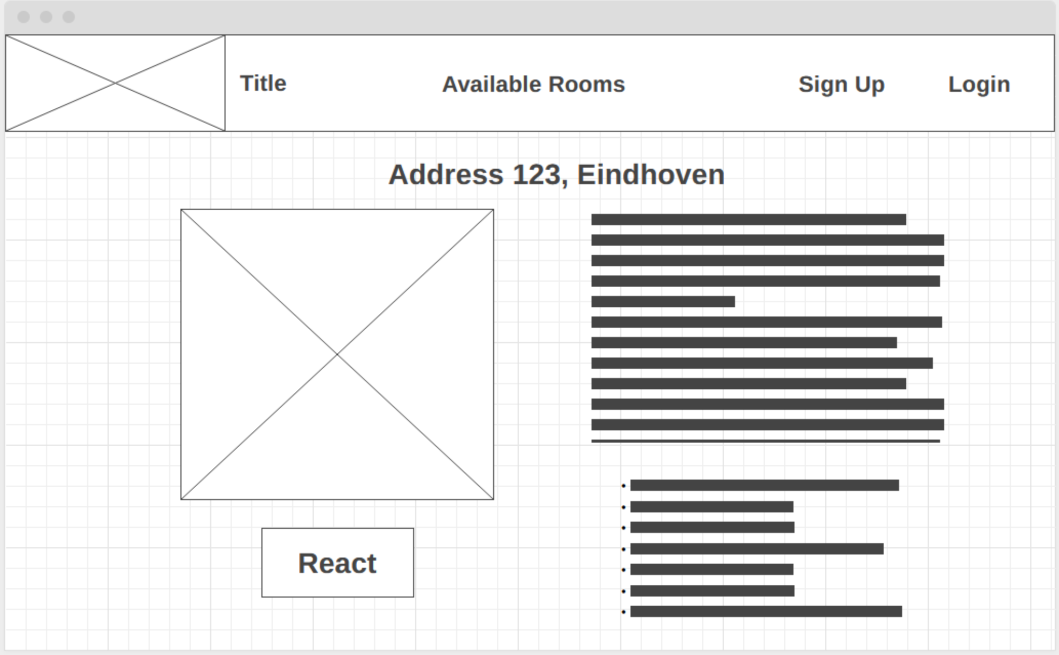
Sprint 3	• Update wireframes • Start work on test plan • Continue implementing US-01 • Implement US-02 1. Create student account classes 2. Create mock data for student accounts 3. Implement respond / react functionality 4. Utilize TDD while writing logic • Style React front-end • Fill in backlog on Jira
Sprint 4	
Sprint 5	

4 Wireframes

4.1 Listings page



4.2 Listing details


Title
Available Rooms
Sign Up
Login
Address 123, Eindhoven
React