1st Sprint – Web Application "Sami Rooms"

Benyamin Yakobi	323492835
Sagi Vaknin	316605617
Liron Vaknin	204447882
Kesem Even-Hen	208055483
Dan Marinescu	204264543

1. Project Documentation

1.1 Functional Requirements

1.1.1 **Website Login Epic** – User shall be able to Log-In as a Student or as a Renter.

User-Stories:

1.1.1.1 As a user, I could login to the website in order to rent\post an apartment.

1.1.1.2 As a user, I can edit my personal information, so that the system will be up to date with my information.

1.1.1.3 As a user, I would like to have option to recover my account password/username in case I have forgot one of the details.

1.1.1.4 As a user, I can open a new renter account, which can post and rent apartments for students.

1.1.1.5 As a user, I can open a student account which allows me to rent apartment during the semester period.

1.1.2 **Student Homepage Epic** – The Student user shall have a Web Page with custom options that specifically customized for Students.

User-Stories:

1.1.2.1 As a user student, I want to be able to log-out after I'm logged in.

1.1.2.2 As a Student, I want to be able to mark a living-unit as an 'Favorite' and my favorites will be displayed in the right side of the screen to watch them later on.

1.1.2.3 As a Student user / visitor of the website, I want to be able to sort living-units by number of rooms, for a convenient display.

1.1.2.4 As a Student user / visitor I want to be able to sort the prices of the living-units that are available from higher to lower and vice versa, for a convenient display.

1.1.2.5 As a Student user / visitor of the website, I want to be able to search living-units by their name, for quick and convenient use.

1.1.2.6 As a Student, I want to see news updates on the front page, so I can keep up with the dorm news.

1.1.2.7 As a Student, I can view all my contracts (Current or old if exist), to manage them.

1.1.2.8 As a Student , I can view all available units that will display each unit information, so that i can easily choose a living unit that matches my conditions.

1.1.2.9 As a student, when I update my personal information, it should be displayed immediately in my student profile, to stay up to date .

1.1.2.10 As a Student, I want to view posts as a table or a gallery, so that i can view all living units easily.

1.1.2.11 As a student, I can see the renter rating and reviews and can write rating and reviews on a renter that I rent apartment from him, to get impression about the renter and to help others.

1.1.3 **Renter Homepage Epic** – The Renter user shall have a Web Page with custom options that specifically customized for Renters.

User-Stories:

1.1.3.1 As a renter, I can tag the living units for its unique properties, to make it more appealing for the students.

1.1.3.2 As a renter, I can review all posted apartments, for easy management and reviewing properties.

1.1.3.3 As a renter, I want to be able to log-out after I am logged in.

1.1.3.4 As a renter, when I update my personal information, it should be displayed immediately in my living-units posts, to stay up to date.

1.1.3.5 As a renter, I can edit information of a living unit that is posted on the website such as price, renting period, etc., so that i could match current real estate market.

1.1.3.6 As a renter, I could post a living unit that i wish to rent which i could enter the min and max renting period, in order that students will rent the unit.

1.1.3.7 As a renter, I want to create or edit a contract, so I can avoid misunderstanding.

1.1.3.8 As a renter, I want option to sort my units by status, so I can view which units has been rented.

1.1.3.9 As a renter, I want to be able to offer discounts by percentages, to attract new customers.

1.1.3.10 As a renter, I want to be updated on each unit change of status, in order to stay up to date with my posted unit.

1.1.3.11 As a renter, I could add photos to posted/posting units, so it could raise the prospects of sale for my unit.

1.1.4 **Booking Management Epic** – Living-Units that are available for rent shall be presented in 'Booking' homepage.

User-Stories:

1.1.4.1 As a user / visitor, I want to be able to see automatic recommendations of outstanding living units based on student rating & their comments.

1.1.4.2 As a user / visitor, I want to be able to see the real price and current price after discount to be able take decision.

1.1.4.3 As a student, I want to be able to leave a comment with my impressions to give feedback to the renter.

1.1.4.4 As a student, I want to be able to rate living-units, to help other students.

1.1.4.5 As a student, I do not want to see apartments which already rented.
1.1.4.6 As a renter, the availability of the unit must be updated after I have rented it, to avoid misunderstanding with other customers.

1.1.5 **Orders Management Epic** – Each user, either Student or Renter, shall be able to view relevant information which relevant to contracts, contact information or information that relevant to a specific order.

User-Stories:

1.1.5.1 as a student, I want to be able to view my Renter information to be able to contact him at any time.
1.1.5.2 As a renter, I want to be able to view my customers (=students) information to be able to contact with them at any time.
1.1.5.3 As a renter I would like to see the status of all my rental units, in order to review my revenue / expenses.
1.1.5.4 As a renter, I should be updated when one of my units sold, so I can make my arrangements for the unit.
1.1.5.5 As a student, I should be able to know when the next due payment, so that i could arrange the money accordingly.

1.1.6 **Payment Window Epic** – Students that have chosen a living-unit to rent, shall be able to go inside a 'Payment' page which includes secure payment methods.

User-Stories:

1.1.6.1 An order page will be displayed containing the condo's information that the user has selected and the final price. The user will enter all personal details and payment methods. If the order details are correct, the order will be saved in the system and sent to the tenant, otherwise an error message will be displayed.
1.1.6.2 As a student, I can choose a payment method for the living unit such as PayPal, visa, credit card etc., so that it will be easier for me to rent a living unit.
1.1.6.3 As a user, I could enter my payment in a well-managed interactive window, so I can pay and view payment details.

1.1.7 **Website Support Epic** – Users that visit or those who signed-up to the Website shall be able to see notifications when the website is down, report bugs and have Customer support by Admins.

User-Stories:

1.1.7.1 as a User, Unless the website is non-operational, the system shall present to users a notification informing the users that the website is unavailable.

1.1.7.2 as an admin, I want to be able to view all tickets properly as a table with the ability to tag each case progress and reply accordingly, in order to give proper customer support.

1.1.7.3 as the Owner , I want to review submitted tickets to the website handle them accordingly, in order to give support to its users.

1.1.7.4 as an Owner, I want an account with full permissions so that I can all the website features and the ability to review tickets, so that I can manage the website more easily and see its progress.

1.1.7.5 As a user, I can report bugs, in order to update the site managers.

1.1.8 **Backend Management Epic** – Whole information shall be saved in databases that have to be updated in a regular basis while the system is online.

User-Stories:

1.1.8.1 As a system admin, I want to backup all databases after a certain amount of time, in order to ensure a fail-safe for the system in case of a corruption in the database.

1.1.8.2 As a system admin, I want to update the Living-Units Database after renter adds/removes a unit to keep the information displayed is up to date.

1.1.8.3 As a system admin, I want to update the Users Database after sign-up process for the user to be able to sign-in the website later / after log-out.

1.1.8.4 As a system admin, I want to update the Living-Units Database after a successful reservation to remove the living-unit from the units that are still displaying for the students.

- Note: **Epics, User-Stories & Tasks** can be found in our ClubHouse!

1.2 Non-Functional Requirements

1.2.1 **Security**

1.2.1.1 Database protection – As we are dealing with client's personal information such as names, addresses, credit cards etc. the database should be secured to prevent from these data to leak.

1.2.1.2 Users must change the permanent login password they receive after password change process immediately after successful login.

1.2.1.3 Password length shall be of 6-10 characters including letters & numbers.

1.2.2 **Performance**

1.2.2.1 Fast I/O – To further user personal experience the system should be able to provide service quickly for the users, Under one second per interaction.

1.2.3 **Scalability**

1.2.3.1 System capacity – The system should be able to work with 500 users simultaneously without crashing.

1.2.4 **Efficiency**

1.2.4.1 The system shall be able to handle the entry of orders by customers at a minimum rate of 10 per second.

1.2.5 **Durability**

1.2.5.1 All databases must be backed up locally to prevent loss of data in case of malicious attack or fatal system crash.

1.2.5.2 The system shall be stored on a well-known & reliable cloud platform to ensure as much as possible its durability.

1.2.6 **Availability**

1.2.6.1 The Online Payment System shall achieve 100 hours MTBF (mean time between failure).

1.2.6.2 The online payment system shall be available 24 hours a day, 7 days a week.

1.2.6.3 Unless the website is non-operational, the system shall present a user with notification informing them that the website is unavailable.

- Note: Those Non-Functional Requirements was mentioned separately to emphasize them. It is important to note that there are Functional Requirements, which had been mentioned above (Section 1.1), that can be interpreted as Non-Functional either.

1.3 Website Login Testcases

1.3.1 **As a user, I could login to the website in order to rent\post an apartment.**

1.3.1.1 Login with the valid credentials i.e. correct username and password.

1.3.1.2 Login with valid username and invalid password.

1.3.1.3 Login with invalid username and valid password.

1.3.1.4 Login with valid username in caps and valid password.

1.3.1.5 Login with invalid credentials, to determine when the account is getting locked, in the sense how many attempts with incorrect credentials are allowed.

1.3.1.6 Login without entering the credentials, by just clicking on the login button to check the validation messages.

1.3.1.7 Login with the valid username and password, observe the URL, make sure the data is not shown in the URL, as it can be preserved and modified later.

1.3.2 **As a user, I can edit my personal information, so that the system will be up to date with my information.**

1.3.2.1 Verify if the user can edit each one of the personal information field.

1.3.2.2 Verify if after pressing edit the field will be available for editing.

1.3.2.3 Verify if the new information that the user enter matches the requirements.

1.3.2.4 Verify if after pressing confirm any changes will be updated in the database.

1.3.2.5 Verify if pressing cancel will discard all changes.

1.3.2.6 Verify if deleting the account also will delete the user from the database.

1.3.2.7 Verify if the user can cancel the action related to deleting the account.

1.3.3 **As a user, I would like to have option to recover my account password/username in case I have forgot one of the details.**

1.3.3.1 To check whether when we select the forgot password link it is directing to forgot password link page.

1.3.3.2 To check whether the link has sent to the mail to which the user has provided.

1.3.3.3 To check whether the user now logins with his password it is working.

1.3.4 **As a user, I can open a new renter account, which can post and rent apartments for students.**

1.3.4.1 Verify that the Registration form contains Username, First Name, Last Name, Password, Confirm Password, Email Id, Phone number, Date of birth, Gender, Location, Terms of use, Submit, Login (If you already have an account).

1.3.4.2 Verify that tab functionality is working properly or not.

1.3.4.3 Verify that Enter/Tab key works as a substitute for the Submit button.
1.3.4.4 Verify that all the fields such as Username, First Name, Last Name, password, and other fields have a valid placeholder.
1.3.4.5 Verify that the labels float upward when the text field is in focus or filled (In case of floating label).
1.3.4.6 Verify that all the required/mandatory fields are marked with * against the field.
1.3.4.7 Verify that clicking on submit button after entering all the mandatory fields, submits the data to the server.
1.3.4.8 Verify that system generates a validation message when clicking on submit button without filling all the mandatory fields.
1.3.4.9 Verify that entering blank spaces on mandatory fields lead to validation error.
1.3.4.10 Verify that clicking on submit button by leaving optional fields, submits the data to the server without any validation error.
1.3.4.11 Verify that case sensitivity of Username (usually Username field should not follow case sensitivity – 'rajkumar' & 'RAJKUMAR' acts same).
1.3.4.12 Verify that system generates a validation message when entering existing username.
1.3.4.13 Verify that the character limit in all the fields (mainly username and password) based on business requirement.
1.3.4.14 Verify that the validation of email field by entering incorrect email id.
1.3.4.15 Verify that the validation of numeric fields by entering alphabets and characters.
1.3.4.16 Verify that leading and trailing spaces are trimmed after clicking on submit button.
1.3.4.17 Verify that the "terms and conditions" checkbox is unselected by default (depends on business logic, it may be selected or unselected).
1.3.4.18 Verify that the validation message is displayed when clicking on submit button without selecting "terms and conditions" checkbox.
1.3.4.19 Verify that the password is in encrypted form when entered.
1.3.4.20 Verify whether the password and confirm password are same or not.
1.3.4.21 Verify if the user cannot proceed without filling all the mandatory fields.
1.3.4.22 Verify if a user can sign-up successfully with all the mandatory details.
1.3.4.23 Verify if the Password field will prompt you for the weak passwords.
1.3.4.24 Verify if duplicate email address will not get assigned.
1.3.4.25 Verify that hints are provided for each field on the form, for the ease of use.

1.3.5 **As a user, I can open a student account which allows me to rent apartment during the semester period.**

1.3.5.1 Verify if the user cannot proceed without filling all the mandatory fields.

1.3.5.2 Verify if the numbers and special characters are not allowed in the First and Last name.

1.3.5.3 Verify if a user can sign-up successfully with all the mandatory details.

1.3.5.4 Verify if the Password and Confirm Password fields are accepting similar strings only.

1.3.5.5 Verify if the Password field will prompt you for the weak passwords.

1.3.5.6 Verify if duplicate email address will not get assigned.

1.3.5.7 Verify that hints are provided for each field on the form, for the ease of use.

1.4 Student Homepage Testcases

1.4.1 **As a user student, I want to be able to log-out after I am logged in.**

1.4.1.1 Verify after successfully login, check logout button is visible or not.

1.4.1.2 Verify by Click on logout s button, after successfully logout on login screen press back button.

1.4.1.3 Verify, login in more than two browser or mobiles and logout from any one from them and check all other account is proper working or all gets logout.

1.4.1.4 Verify all UI components are visible properly.

1.4.1.5 Logout option should not be visible till the user is logged in.

1.4.1.6 Verify the spelling of Logout option.

1.4.1.7 Verify that link of logout is clickable when a user is logged in.

1.4.2 **As a Student, I want to be able to mark a living-unit as a 'Favorite' and my favorites will be displayed in the right side of the screen to watch them later.**

1.4.2.1 Verify if there is a favorite button next to each unit.

1.4.2.2 Verify if Uncheck the favorite button will remove the unit from favorite database.

1.4.2.3 Verify if each student will have his own favorite units page that will present only the units that has been checked as favorite by the student, also if the student log out.

1.4.3 **As a Student user / visitor of the website, I want to be able to sort living-units by number of rooms, for a convenient display.**

1.4.3.1 Verify if sort ability sorts the number of rooms of the living-units that are available from higher to lower.

1.4.3.2 Verify if sort ability sorts the number of rooms of the living-units that are available from lower to higher.

1.4.3.3 Verify if sort by number of rooms button works.

1.4.4 **As a Student user / visitor I want to be able to sort the prices of the living-units that are available from higher to lower and vice versa, for a convenient display.**

1.4.4.1 Verify if sort ability sorts the prices of the living-units that are available from higher to lower.

1.4.4.2 Verify if sort ability sorts the prices of the living-units that are available from lower to higher.

1.4.4.3 Verify if sort by price button works.

1.4.5 **As a Student user / visitor of the website, I want to be able to search living-units by their name, for quick and convenient use.**

1.4.5.1 Verify that after entering data to the search bar and pressing search button, any matches that has been found between the search data and unit information will be displayed.

1.4.5.2 Check cursor position of the search box when typing starts.

1.4.5.3 Have commissions for changing the color of visited pages or links. Check that the color varies for the visited ones and remains unchanged for new links.

1.4.5.4 Check the search response time. Ensure that it is up to standard and not too long.

1.4.5.5 Check that no results will be displayed on single character searches that might occur by mistake.

1.4.5.6 Search results displayed should be relevant to search keyword.

1.4.5.7 % sign in search keyword should not redirect to 404 ERROR.

1.4.5.8 Application should not crash if user inserted % in search field.

1.4.5.9 When user start typing word in text box it should suggest words that matches typed keyword.

1.4.5.10 After clicking Search field - search history should be displayed (latest search keyword).

1.4.5.11 All search keyword/filters should get cleared on clicking Reset button.

1.4.5.12 Search results should be cleared on clicking clear search button.

1.4.5.13 History displayed in search field should be relevant to logged in user only.

1.4.5.14 Total number of search records/results should be displayed on page.

1.4.5.15 User should be able to search when he enters the keyword and hits 'Enter' button on keyboard.

1.4.6 **As a Student, I want to see news updates on the front page, so I can keep up with the dorm news.**

1.4.6.1 Verify if there are news update on front page that will display all new messages.

1.4.6.2 Verify if there is effect that makes the news scroll down.

1.4.7 **As a Student, I can view all my contracts (Current or old if exist), to manage them.**

1.4.7.1 Verify if the "view contracts" button will be displayed on the student homepage.

1.4.7.2 Verify if pressing the "view contract" button will open the renter contract for the apartment.

1.4.7.3 Verify if the "view contracts" button will be displayed on the student homepage.

1.4.7.4 Verify if pressing the "view contract" button will open the renter contract for the apartment.

1.4.8 **As a Student , I can view all available units that will display each unit information, so that i can easily choose a living unit that matches my conditions.**

1.4.8.1 Verify if the system will present all the posted apartments.

1.4.8.2 Verify if the apartment filtering ability by location, start or end of rental, rental price, renter name, status works.

1.4.9 **As a student, when I update my personal information, it should be displayed immediately in my student profile, to stay up to date.**

1.4.9.1 Verify if the information has been entered correctly.

1.4.9.2 Verify if after changing personal information it will be update in every unit that the account posted in the database accordingly.

1.4.9.3 Verify if after changing personal information it will be update in the database.

1.4.10 **As a Student, I want to view posts as a table or a gallery, so that i can view all living units easily.**

1.4.10.1 Verify if after pressing 'gallery' button all units will be shown as pictures.

1.4.10.2 Verify if after pressing 'table' button all units will be shown by details.

1.4.11 **As a student, I can see the renter rating and reviews and can write rating and reviews on a renter that I rent apartment from him, to get impression about the renter and to help others.**

1.4.11.1 Verify if the confirmation button, after entering review this button will send the review to the server.

1.4.11.2 Verify if when a living unit will be selected, there is a button that will show text box and rating scale for the user to enter his review.

1.4.11.3 Verify if when a living unit will be selected, it's shows the renter's rating and reviews.

1.5 Renter Homepage Testcases

1.5.1 **As a renter, I can tag the living units for its unique properties, to make it more appealing for the students.**

1.5.1.1 Verify that tagging system added to each unit.

1.5.1.2 Verify adding a tag is saved into the unit and the database matching.

1.5.1.3 Verify that when editing/creating a unit a button shows in order to add tags.

1.5.2 **As a renter, I can review all posted apartments, for easy management and reviewing properties.**

1.5.2.1 Verify if the system will present page of all the posted apartments by other renters.

1.5.2.2 Verify if the system will present page of all the renter apartments.

1.5.2.3 Verify if the apartment filtering ability by location, start or end of rental, rental price, status works.

1.5.2.4 Verify if from the homepage the click of edit all his apartments detail is open a new window.

1.5.2.5 Verify if from the homepage the click of add new apartment is open a new window.

1.5.2.6 Verify if from the homepage the click of 'My contract' open this page.

1.5.3 **As a renter, I want to be able to log-out after I am logged in.**

1.5.3.1 Verify after successfully login, check logout button is visible or not.

1.5.3.2 Verify by Click on logout s button, after successfully logout on login screen press back button.

1.5.3.3 Verify, login in more than two browser or mobiles and logout from any one from them and check all other account is proper working or all gets logout.

1.5.3.4 Verify all UI components are visible properly.

1.5.3.5 Logout option should not be visible till the user is logged in.

1.5.3.6 Verify the spelling of Logout option.

1.5.3.7 Verify that link of logout is clickable when a user is logged in.

1.5.4 **As a renter, when I update my personal information, it should be displayed immediately in my living-units posts, to stay up to date.**

1.5.4.1 Verify if the information has been entered correctly.

1.5.4.2 Verify if after changing personal information it will be update in every unit that the account posted in the database accordingly.

1.5.4.3 Verify if after changing personal information it will be update in the database.

1.5.5 **As a renter, I can edit information of a living unit that is posted on the website such as price, renting period, etc., so that i could match current real estate market.**

1.5.5.1 Verify if for renter account when selecting a unit, the button for editing the selected unit open page for editing the post.

1.5.5.2 Verify if after the submissions of the changes, the button will show up that will update the system of the changes in the unit.

1.5.5.3 Verify if the button for saving the form works only after all the files are full as it should be.

1.5.5.4 Verify if the changes will update in the database.

1.5.6 **As a renter, I could post a living unit that i wish to rent which i could enter the min and max renting period, in order that students will rent the unit.**

1.5.6.1 Verify if the unit information form will include the fields: location (city, address), price, min and max renting period, description.

1.5.6.2 Verify if the button that will open a submission form for entering unit information works.

1.5.6.3 Verify if the button for saving the form and sending it to the server which will insert it into the database and add it as a unit.

1.5.6.4 Verify if the button for saving the form works only after all the files are full as it should be.

1.5.7 **As a renter, I want to create or edit a contract, so I can avoid misunderstanding.**

1.5.7.1 Verify if the button to open submission form of a contract works.

1.5.7.2 Verify if the button to editing an existing contract form works.

1.5.7.3 Verify if after the submissions of changes, there is a button that show and will update the system of the changes in the contract.

1.5.7.4 Verify if the changes will send to the tenant.

1.5.7.5 Verify if the changes will update in the database.

1.5.8 **As a renter, I want option to sort my units by status, so I can view which units has been rented.**

1.5.8.1 Verify if the sort by status shows: rented, offered for rent, In the rental process.

1.5.8.2 Verify is the sort display the correct apartments.

1.5.9 **As a renter, I want to be able to offer discounts by percentages, to attract new customers.**

1.5.9.1 Verify if you can add a valid discount.

1.5.9.2 Verify if you can add an invalid discount.

1.5.9.3 Verify if the discount feature, will show the original price and the discounted one.

1.5.9.4 Verify that the new price update in the database.

1.5.10 **As a renter, I want to be updated on each unit change of status, in order to stay up to date with my posted unit.**

1.5.10.1 Verify after any status change, that the renter user Is getting a notification update.

1.5.10.2 Verify that the notification update text shows the correct information of the update.

1.5.10.3 Verify that each notification updated is saved into the database.

1.5.11 **As a renter, I could add photos to posted/posting units, so it could raise the prospects of sale for my unit.**

1.5.11.1 Verify the button exists and functions.

1.5.11.2 Verify the file opener works and uploads the photo.

1.5.11.3 Verify the drag option uploads the photo.

1.5.11.4 Verify the confirm button works, and that the photo is saved into the database.

1.6 Booking Management Testcases

1.6.1 **As a user / visitor, I want to be able to see automatic recommendations of outstanding living units based on student rating & their comments.**

1.6.1.1 Verify there is an available unit that also has recommendation

1.6.1.2 Verify the unit above is shown in the menu

1.6.1.3 Verify the unit rating and comments of other students is shown.

1.6.1.4 Verify that all the units that shown is supposed to be in the recommendation lists.

1.6.2 **As a user / visitor, I want to be able to see the real price and current price after discount to be able take decision.**

1.6.2.1 Verify there is available unit for rent.

1.6.2.2 Verify the unit information appears

1.6.2.3 Verify the price before and after the discount is shown.

1.6.2.4 Verify the per-cents discount shown

1.6.3 **As a student, I want to be able to leave a comment with my impressions to give feedback to the renter.**

1.6.3.1 Verify that the comment section is available.

1.6.3.2 Verify that the message sender has received feedback if the message has sent successfully.

1.6.3.3 Verify the renter receive the message

1.6.4 **As a student, I want to be able to rate living-units, to help other students.**

1.6.4.1 Verify that there is living unit post.

1.6.4.2 Verify the rate option is available.

1.6.4.3 After a rate, the total rating should update

1.6.4.4 After rating again, the previous rate would be deleted, and the new rate should appear.

1.6.5 **As a student, I do not want to see apartments which already rented.**

1.6.5.1 Verify that there is unit that already rented in the database.

1.6.5.2 Verify that the rented unit does not shown in the available database.

1.6.5.3 Verify that the rented unit does shown in the unavailable database.

1.6.5.4 Verify the rented unit from above does not appear on the search options.

1.6.5.5 Verify that the unit that chosen is available for renting.

1.6.6 **As a renter, the availability of the unit must be updated after I have rented it, to avoid misunderstanding with other customers.**

1.6.6.1 Verify that a unit from specific renter has been rented.

1.6.6.2 Verify that the unit from above shows in the rented database units

1.6.6.3 Verify that the unit from above does not show in the available units.

1.6.6.4 Verify that the rented unit does not show in the search options.

1.7 Orders Management Testcases

1.7.1 **as a student, I want to be able to view my Renter information to be able to contact him at any time.**

1.7.1.1 Verify the renter contact button is shown in the webpage.

1.7.1.2 Verify that the button redirects into a matching webpage that includes the current renter for the active contract with the logged user.

1.7.1.3 Verify that the renter credentials and info match the database.

1.7.1.4 Verify the contact information is correct and matches the database.

1.7.1.5 Verify that all the contact renter options are available and are working.

1.7.2 **As a renter, I want to be able to view my customers (=students) information to be able to contact with them at any time.**

1.7.2.1 Verify the students contacts button is shown in the webpage.

1.7.2.2 Verify that the button redirects into a matching webpage that includes all the contacts.

1.7.2.3 Verify that each contact that is shown is in an active renting period with the current renter account.

1.7.2.4 Verify the contact information is correct and match the database.

1.7.2.5 Verify that a student contact can be selected.

1.7.2.6 Verify when student contact selected that all the contact user options are available and are working.

1.7.3 **As a renter I would like to see the status of all my rental units, in order to review my revenue / expenses.**

1.7.3.1 Verify that the button of renting total status is shown in homepage.

1.7.3.2 Verify that the button redirects into a matching webpage that includes all the data.

1.7.3.3 Verify that the shown status for each unit match the data in the database.

1.7.3.4 Verify for each calculation that it is correct.

1.7.4 **As a renter, I should be updated when one of my units sold, so I can make my arrangements for the unit.**

1.7.4.1 Verify when a sell is confirmed on the server side, It also activates a function to notify the renter.

1.7.4.2 Verify as a renting account that have sold unit, that it is notified of the sale.

1.7.4.3 Verify when a notification of a sale is pressed, it redirects to the page with all the selling information and the contract.

1.7.5 **As a student, I should be able to know when the next due payment, so that i could arrange the money accordingly.**

1.7.5.1 Verify when rent payment is due, an event on the server side occurs and sends it the renter.

1.7.5.2 Verify as a student account that the notification received with all the details.

1.7.5.3 Verify when a payment notification is pressed, it will redirect to a payment method page.

1.8 Payment Window Testcases

1.8.1 **An order page will be displayed containing the condo's information that the user has selected and the final price. The user will enter all personal details and payment methods. If the order details are correct, the order will be saved in the system and sent to the tenant, otherwise an error message will be displayed.**

1.8.1.1 Verify that a unit has been selected.

1.8.1.2 Verify that a unit is available for rent.

1.8.1.3 Verify that all the unit price and other details is shown.

1.8.1.4 Verify all the student personal details and payment methods are correct.

1.8.1.5 Verify that invalid information display error message.

1.8.1.6 Verify that correct order saved in the system.

1.8.1.7 Verify that order is sent to tenant.

1.8.2 **As a student, I can choose a payment method for the living unit such as PayPal, visa, credit card etc., so that it will be easier for me to rent a living unit.**

1.8.2.1 Verify that payment methods is shown.

1.8.2.2 Verify that the information change by the payment methods (e.g. PayPal fields are different than visa fields).

1.8.2.3 Verify that all payment methods that offered are displayed.

1.8.3 **As a user, I could enter my payment in a well-managed interactive window, so I can pay and view payment details.**

1.8.3.1 Verify that after entering payment page the interactive window is shown.

1.8.3.2 Verify that all the payment details are shown.

1.8.3.3 Verify that successful payment has been transfer to bank account.

1.9 Website Support Testcases

1.9.1 **as a User, Unless the website is non-operational, the system shall present to users a notification informing the users that the website is unavailable.**

1.9.1.1 Verify the website shut down option available.

1.9.1.2 Verify the website still down after choosing to.

1.9.1.3 Verify the website is up again.

1.9.1.4 Verify the website is still up after choosing to.

1.9.1.5 Verify that users that tries to log in receive a notification that the website is unavailable.

1.9.1.6 Verify that users that already logged in receives a message that the website is unavailable.

1.9.2 **as an admin, I want to be able to view all tickets properly as a table with the ability to tag each case progress and reply accordingly, in order to give proper customer support.**

1.9.2.1 Verify that you have admin permissions.

1.9.2.2 Verify that you can view table of tickets.

1.9.2.3 Verify that you can tag cases which appears in the table.

1.9.2.4 Verify that you can reply to customer which reported a case that appears in the table.

1.9.3 **as the Owner , I want to review submitted tickets to the website handle them accordingly, in order to give support to its users.**

1.9.3.1 Verified tickets in the website.

1.9.3.2 Verify that the order information is available

1.9.4 **as an Owner, I want an account with full permissions so that I can all the website features and the ability to review tickets, so that I can manage the website more easily and see its progress.**

1.9.4.1 Verify that admin account has access to all permissions.

1.9.4.2 Verify admin account can connect and authorize with the system.

1.9.4.3 Verify after login that it is redirects to the matching page.

1.9.4.4 Verify that as an admin account it can change profile (such as a renter, student, admin).

1.9.5 **As a user, I can report bugs, in order to update the site managers.**

1.9.5.1 Verify the report a bug button available.

1.9.5.2 Verify after pressing the button a report shows up.

1.9.5.3 Verify the bug report appears in the list of the bugs.

1.10 Backend Management Testcases

1.10.1 **As a system admin, I want to backup all databases after a certain amount of time, in order to ensure a fail-safe for the system in case of a corruption in the database.**

1.10.1.1 Verify that data is saved to the backup database after the defined amount of time.

1.10.2 **As a system admin, I want to update the Living-Units Database after renter adds/removes a unit to keep the information displayed is up to date.**

1.10.2.1 Verify that after adding new unit it appears on the database.
1.10.2.2 Verify that the new unit is available in the search options.

1.10.3 **As a system admin, I want to update the Users Database after sign-up process for the user to be able to sign-in the website later / after log-out.**

1.10.3.1 Verify that after new sign-up new user appears on the database.
1.10.3.2 Verify that new users can log in/log out.

1.10.4 **As a system admin, I want to update the Living-Units Database after a successful reservation to remove the living-unit from the units that are still displaying for the students.**

1.10.4.1 Verify that after adding new unit it appears on the database.
1.10.4.2 Verify that the new unit is available in the search options.

2. Project Infrastructure

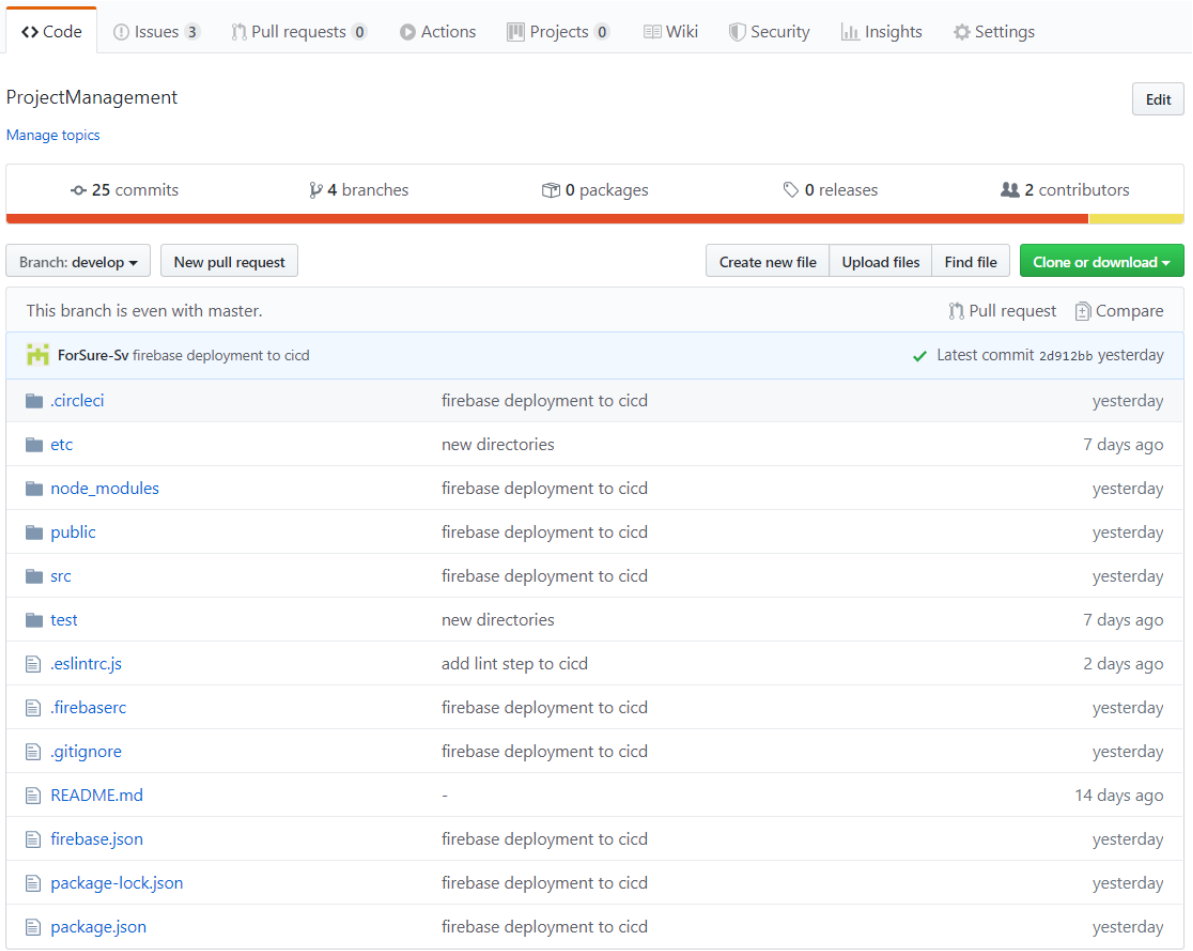
2.1 Conventions – In this project we will use these conventions:

2.1.1 The 'camelCase' Convention.

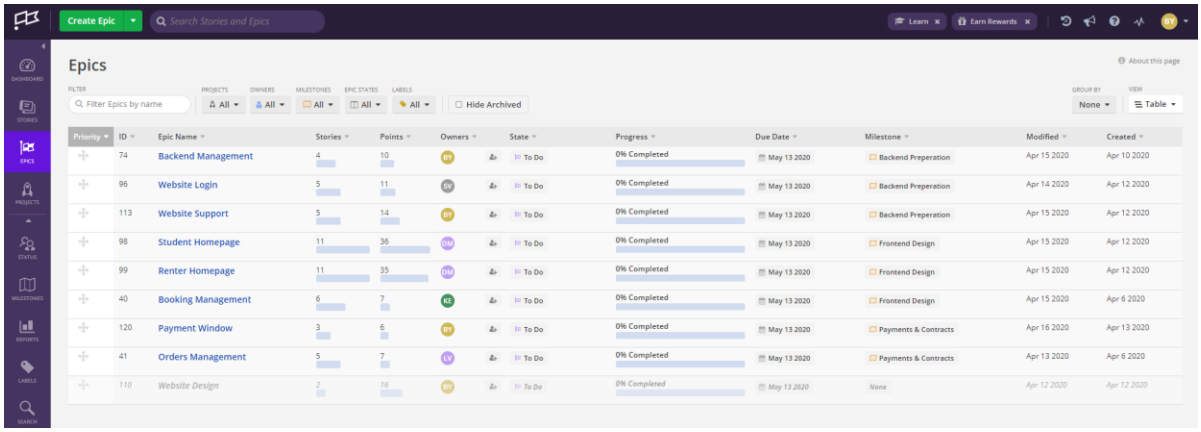
2.1.2 The 'Curly Brace' Convention.

2.1.3 The 'beforeBlockComment' Convention.

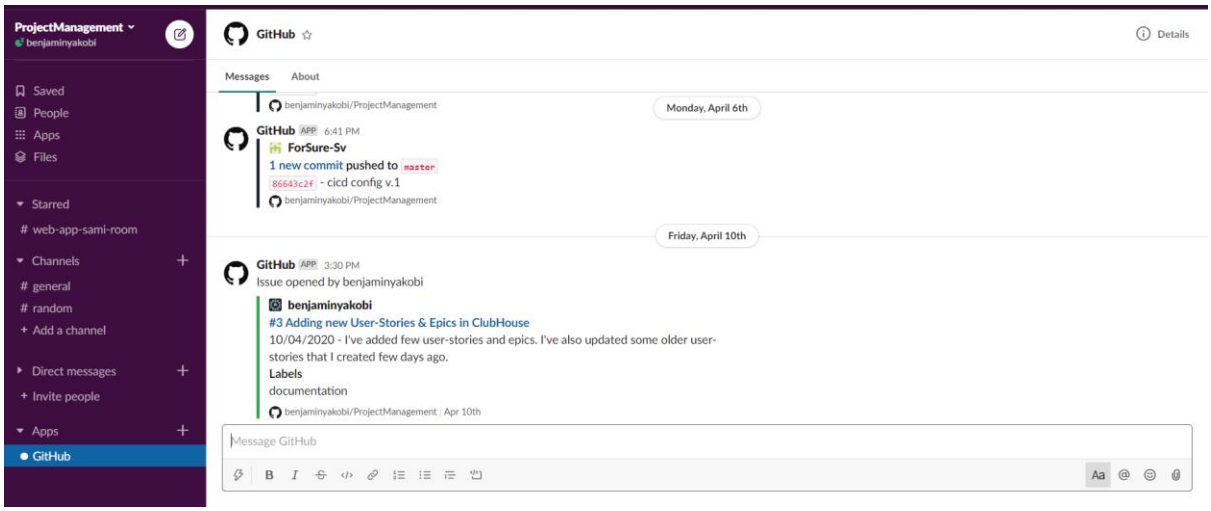
2.2 GitHub Repository & Screenshot:



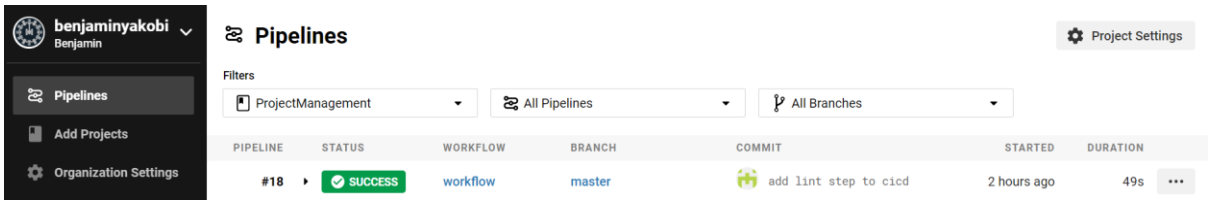
2.3 Clubhouse Invitation & Screenshot:



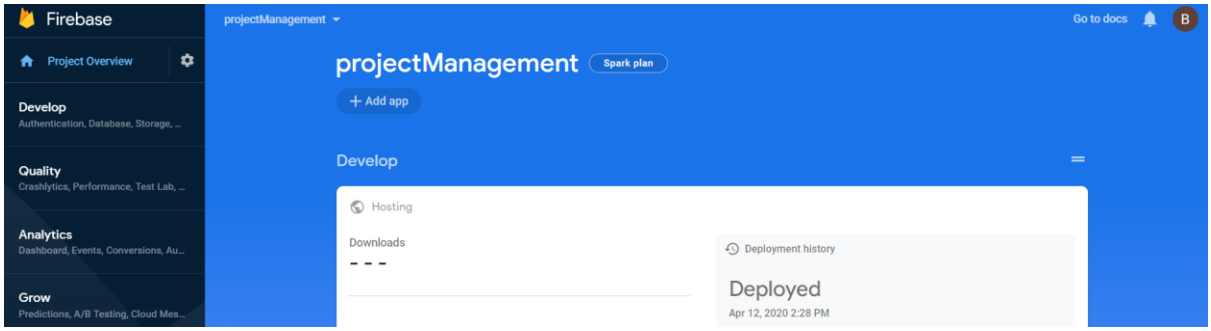
2.4 Slack Invitation & Screenshot:



2.5 CircleCi & Screenshot:



2.6 Firebase & Screenshot:



- In this project we will use Firebase as Cloud service to store & run the project.
- In this project we will use Firebase as our DB to save our data for the next sprints.

2.7 Sprint Cycle:

2.7.1 Development process of the product will take few short sprints until accomplishment, while the focus on current sprint was on creating a solid foundation for the next sprints. In the next sprints the product will be written mainly in JavaScript language with emphasis on using HTML & CSS as well, to improve visibility of the final product.