

F29SO – Software Engineering  
Individual Coursework (CW1) – Calum Murray  
**Airbnb**  
([www.airbnb.com](http://www.airbnb.com))

**Brief**

This report specifies a requirements engineering process and group management strategy for the development of an Airbnb holiday rental system [www.airbnb.com](http://www.airbnb.com)

The function of this document is to demonstrate my competency and understanding of group management, software engineering, functional/non-functional requirements and requirements prioritization.

**Functional Requirements**

**F-UR1: User/Account Management (Priority: MUST)**

1. Account Creation and Management
  1. Users must be able to create accounts, edit their profiles, and manage their account details.
  2. The system must support two primary user roles: Guests and Hosts
  3. Users should have the ability to reset their passwords
2. Role Dependent Access
  1. Access to system features should be dependent on the type of user: Guest and Host

**F-UR2: Property Listing Management (Priority: MUST)**

1. Listing Creation and Management
  1. Hosts must be able to create, edit and remove property listings
  2. Listings must contain details about the property such as: description, location, services, price, availability, etc.
  3. Hosts should have the capacity to upload images of the property
2. Calendar Interface
  1. Host Calendar View
    1. Hosts should be provided with a calendar interface to manage property availability
  2. Guest Calendar View
    1. Guests should be able to view the calendar interface to view availability and book for certain dates

**F-UR3: Listing Search (Priority: MUST)**

1. Property Listing Search
  1. Guests must be able to search for listings by properties such as location, property types, price range, etc.
  2. Guests should be given more comprehensive filtering and sorting criteria for their results

**F-UR4: Booking Management (Priority: MUST)**

1. Availability Viewing
  1. Guests must be able to view property availability for certain dates
2. Guest Booking Request
  1. Guests must be able to request booking for available dates
3. Host Booking Request Management
  1. Hosts should be able to accept/reject incoming booking requests
  2. Hosts should be able to update listing availability

**F-UR5: Message System (Priority: SHOULD)**

1. Guest-Host Communication
  1. Guests and Hosts should be able to message one another to facilitate inquiries and coordinate details

**F-UR6: Review/Feedback System (Priority: SHOULD)**

1. Guest Reviews
  1. Guest should be able to leave reviews on property listings
2. Host Feedback
  1. Hosts should be able to review guests, to assist other hosts when they review booking requests

**F-UR7: Customer Support (Priority: COULD)**

1. FAQ Page
  1. A section with Frequently Asked Questions and other helpful information
2. Support Chat
  1. Users could submit support tickets to request a message interface with support staff

**F-SR1: Payment Processing (Priority: MUST)**

1. The system must provide payment processing for bookings

**F-SR2: GPS Integration (Priority: COULD)**

1. The system could integrate with third party mapping/GPS systems such as Google Maps
2. This could include navigation support for guests

**Non-Functional Requirements****NF-UR1: Usability (Priority: SHOULD)**

1. UI Design
  1. The system should have an intuitive and appealing UI, especially for guests

**NF-UR2: Localization (Priority: COULD)**

1. Multi-Language Support
  1. The system could support multiple languages for global users and tourists
2. Currency Support
  1. The system could recognise, based on user location, what currency to display
  2. The system could handle currency conversion

**NF-SR1: Performance (Priority: MUST)**

1. Minimize Response Time
  1. The system must maintain low latency and load time

**NF-SR2: Security (Priority: MUST)**

1. Data Security
  1. Sensitive data must be encrypted
2. User Authentication
  1. Must include secure log-in mechanisms like two-factor authentication

**NF-SR3: Reliability (Priority: MUST)**

1. Service Availability
  1. The system must ensure downtime is minimized
2. Data Accuracy
  1. Data stored should be backed up
  2. The system should routinely check that data is up-to-date and accurate
  3. The system must identify if data is corrupted and roll back to an accurate state if possible

**NF-SR4: Legal Compliance (Priority: MUST)**

1. The system must comply with GDPR and other relevant laws

**NF-SR5: Scalability (Priority: SHOULD)**

1. The system should support scaling of the data storage system to accommodate increased number of users and listings
2. The system should support scaling of the network capacity to accommodate increased traffic

**NF-SR6: Compatibility (Priority: SHOULD)**

1. The system should be compatible with different platforms and browsers

**Requirements Engineering Process****Feasibility Study:**

A feasibility study analyses a projects likelihood to succeed and evaluates its practicality. Considering the context of this report, it was not necessary for me to conduct this stage.

**Requirements Elicitation:**

This is the process of gathering requirements from users/stakeholders/clients/etc., which normally involves actively engaging with these individuals/organisations to ensure that the system aligns with the requirements of the stakeholders.

As I do not have access to stakeholders, the only method I have used to elicit requirements is analysis and observation of the Airbnb website, as well as a couple of competitors:

[www.airbnb.com](http://www.airbnb.com), [www.booking.com](http://www.booking.com), [www.tripadvisor.com](http://www.tripadvisor.com);

This process gathered an informal and unstructured list of functional and non-functional requirements that I could further analyse and formalize in the next steps of the requirements engineering process.

### Requirements Analysis:

Analysis is the step in which the requirements gathered during elicitation are refined and organised. This is to ensure that the gathered wants and needs of stakeholders are filtered and formatted to allow engineers to have a more practical list of objectives and criteria to meet.

Requirements prioritization is the process of identifying and ranking the importance of gathered requirements to assist in efficient allocation of time and resources to the most critical requirements. The method that I implemented is the MoSCoW method, in which requirements are labelled:

- MUST – Essential to core functioning of the system
- SHOULD – Important, but can be delayed
- COULD – Not important, but can be implemented if possible within project constraints
- WON'T – Not within the remit of the project

(note: because of the nature of my elicitation method, it did not find it reasonable to gather any WON'T requirements)

### Requirements Specification:

This is where the requirements gathered are formally specified and documented. This provides a comprehensive description of what is expected to be produced. The documentation will serve as an essential reference for the client, developers, testers, project manager etc., throughout the whole lifecycle of the project.

This is the stage where I produced and labelled the final list of functional and non-functional requirements for this report.

### Requirements Validation:

This is the process of ensuring that the documented requirements accurately represent the expectations/needs of the stakeholders. This is important to ensure that development doesn't begin on aspects of the project that end up being a invalid, which may be costly to correct.

This is also an opportunity to speculate project costs.

As the scope of this report doesn't include interaction with the stakeholders, the relevant method that I used in this stage was to cross-reference my requirements with the Airbnb website and relevant systems. This aimed to align my proposed system with established systems.

## **Group Management Approach and Team Structure**

### **Team Structure**

To build this system, I would assign the following roles:

- Project Manager
  - The overall manager and coordinator of the project, their job is to ensure the system fulfils the requirements and meets the clients needs
- Coordinator
  - Coordinates the project, their job is to manage resources, track progress and keep the team organised
- Technical Engineer
  - They are in charge of technical decisions, should be familiar with the technology the system is built on/with
- Frontend Developer
  - They are responsible for the parts of the system that the users interact with, UI/UX
- Backend Developer
  - They are responsible for server side implementation, such as data storage, transactions and accounts
- Quality Assurance
  - They are in charge of ensuring that bugs and faults in the system are identified and fixed

### **Group Management Approach**

To maintain efficient teamwork, it is vital to ensure regular communication, therefore regular meetings and communication tools will be implemented. A system for team members to provide feedback will be provided.

## **Software Design Methodology**

For the development of this system, I would implement an Agile Incremental methodology. This allows for flexibility, as it will build the system in small increments. The small segmentation of work allows for regular feedback and review.