Software Design Document

<Sydney Airbnb>

Chelzie Castanares – s5259144

Thien Thao My Bui -

Table of Contents

[1.0 System Vision 3](#_Toc46748622)

[1.1 Problem Background 3](#_Toc46748623)

[1.2 System Overview 3](#_Toc46748624)

[1.3 Potential Benefits 3](#_Toc46748625)

[2.0 Requirements 4](#_Toc46748626)

[2.1 User Requirements 4](#_Toc46748627)

[2.2 Software Requirements 4](#_Toc46748628)

[2.3 Use Cases 4](#_Toc46748629)

[3.0 System Components and Software Design 5](#_Toc46748630)

[3.1 System Components 5](#_Toc46748631)

[3.2 Software Design 5](#_Toc46748632)

[4.0 User Interface Design 6](#_Toc46748633)

# System Vision

## Problem Background

Airbnb is a global business that specialises in providing accommodation but differs from traditional hotels and resorts. Airbnb acts a marketplace between homeowners who wish to rent out their place for accommodation and tourists looking for a place to stay. As the popularity of Airbnb increases worldwide, the need of updated systems is more prevalent within their business. Sydney Airbnb is one of the most popularized tourist destinations within the world ranking at 4th in 2016 and therefore has been chosen as the subject to perform data analysis using a visualization tool to understand and improve the business structure to increase efficiency.

* Problem of How to use data

## System Overview

The system should be able to perform a range of capabilities that’ll enable Airbnb to analyse and produce results from Sydney’s data.

The following include the system capabilities identified:

* Within a user-selected period, return all the listings in a specified suburb.
* Within a user-selected period, display the distribution of prices of properties using a chart.
* Within a user-selected period, retrieve all records that contain a specific keyword (e.g., pool, 2-bedroom).
* Analyse the comments relating to cleanliness of the property such as (dust, dirt etc.).
* Return all the reviews of a user-selected property.
* What the system does, how it access data
* Solution overview

## Potential Benefits

With the use of the of the data visualisation tool, it will help Airbnb understand the data behind Sydney’s traffic movement which help in turn helps the business to understand its consumers better. The following system will allow the business to analyse popular locations of renting Airbnb’s and how well the customers enjoyed their stay based on reviews left. This will enable the business to produce more value for consumers as it aims to improve problems based on data analysed.

The potential benefits include but are not limited to:

* Increased customer satisfaction: Consumers are more likely to increase due to the improvement of feedbacks thy have given.
* Increased efficiency: The use of the data visualisation tool makes it easier for the business to identify problems and tackle issues.
* Improved cost:

# Requirements

## User Requirements

In this section you detail how a user is supposed to interact with or use your program. What do they ***need*** to be able to do? This should all be from the end users perspective. Can be a combination of narrative text and listing of needs.

**Assignment note: You have not been given a client/user, so you can make one up. Who do you think would be using your software?**

Our client Airbnb has listed specific requirements to be included. The following system should include the following user requirements that enable the business to work on the issue.

|  |  |
| --- | --- |
| Requirements | Description |
| Property Search | The user is able to input the location of stay within their selected period date. |
| Property Price Search | The user is able to see the cost of property within their selected period date. |
| Keyword Search System | The user is able to search for properties containing the specific keyword/s they are looking for. |
| Comments Search System | The user is able to view all comments of selected property within their selected period date. |
| Review System | The user is able to look at the reviews of a certain property. |

## Software Requirements

In this section you detail what the requirements for the software are. What functionality will it provide? This is usually a formal listing, with requirements often using the word ‘Shall’. IE:

R1.1 The program shall accept multiple file names as arguments from the command line.

R1.2 Each file name can be a simple file name or include the full path of the file with one or more levels.

etc …

Can be primarily functional requirements, though you may include other types if you think of them.

The following are non-functional requirements that focus on FURPS+ (Functionality, Usability, Reliability, Performance and Security):

|  |  |
| --- | --- |
| Non - Functional Requirements | |
| FURPS+ Category | Description |
| Usability | * User interface should be easy to use and navigate. * All requirements should be working smoothly with no errors. |
| Reliability | * System should be running at all times. * System should be error free. |
| Performance | * System response time should be quick when retrieving data e.g., 5 secs. * Should be able to store large amounts of data. |
| Security | * Website should be encrypted with HTTPS Protocol. |

## Use Cases & Use Case Diagrams

In this section you provide some use cases showing how people may use your software

|  |  |
| --- | --- |
| **Use Case ID** |  |
| **Use Case Name** |  |
| **Actors** |  |
| **Description** |  |
| **Flow of events** |  |
| **Alternative Flow** |  |

# Software Design and System Components

## Software Design

A block diagram/flowchart of how your software might work

## System Components

### Functions

Preliminary list of all functions in the software. For each function in the list the following information is provided:

* a brief description of what it does (1 or 2 sentences);
* a list of the input parameters, and their data types, and what they are used for;
* a list of any side effects caused by the function (ie change global or member variables, changes data passed by reference from calling function etc)
* a description of the function’s return value

### Data Structures / Data Sources

List of all data structures in the software (eg linked lists, trees, arrays etc) or eternal data sources. For each data structure in the list the following information is provided:

* Type of structure (tree, list etc),
* Description of where and how it is used
* List of data members, and what each one is for do
* List of functions that use it

### Detailed Design

Pseudocode for all non-standard / non-trivial algorithms that operate on data structures

Commenst of data = E.g get data

# User Interface Design

This is your initial interface design. Describe the tools you used for this design stage and any key findings that informed your design. This introduction is descriptive and should explain what you have completed for the actual design work you will present in the sub-sections below.

## Structural Design

Structural design refers to the navigational and information structure of your product – the structure that supports the interface layout. How will you structure your product? How will you group your information? How will you navigate through your product? Why? This can take the form of a diagram showing structure and hierarchy, supported by a discussion and justification of your choices. Why have you made these design choices? Describe and outline the structure of your interface and of your information.

## Visual Design

Detail your visual design: Layout, visual elements, icons, graphics, style, colour, fonts general screen designs. This can be sketches, wireframes, mockups etc, supported by a discussion, explanation, and justification of your choices.