

**Accommodation Sharing System**

# A project work for CSE470

# **Introduction**

Accommodation Sharing System will be an organization that will enable house/apartment owners to share their apartment/house with visitors/accommodation seekers in exchange for payment for the days the visitors will be staying. The system will also have options for home/apartment owners to advertise their house/apartment for rent. The owners will post descriptions about their home/apartments, pictures, facilities provided by them. People can post paid advertisements of other stuffs. The owners and home seekers both need to register first by creating an account which will require username and password. The system will have a search option to look for homes/apartments based on bachelors wanted or not, smoking restrictions, room sizes, accommodation facilities like internet, dining, 3 meals a day, hot water, television and other amenities. The owners will be able to review all the booking requests from rentee’s and select a rentee according to the liking of their choice.

# **Motivation**

Getting a place like home to live for a few days can be very difficult to find especially for someone visiting for a few days or a bachelor or student attending at a University far from home. Hotels can be a solution but the cost of living becomes high and they are not exactly a living home for everyone. Even though there are apartment for rents most of the owners don’t rent out to bachelors, therefore mostly students and bachelors face problems finding places to live. Furthermore, it is difficult to find houses for rent manually going from one are to another. Most of the time the landlord or property manager is not available on visit. On the other end house owners also have to go through the tedious process of reviewing rent applicants and verifying them, also finding a matching renter according to their choice takes a lot of time. Therefore, this system proposal was initiated where home owners with extra rooms or extra apartments can share their houses/apartments with different home like facilities in exchange for rent.

# **Methodology**

The term methodology means the technique and procedure adopted by conducting a research study. It outlines how data will be collected and the tools for collecting data, system methodology, the proposed system input and output, users and systems development tools. In our system we will be using **agile development** which is much efficient code-based process. Our main purpose will be to develop a correct system which does not require extensive documentation and to be able to respond quickly to changing requirements without excessive rework. In our system project manager will value individual interactions over systems and tools. Moreover, we will be focusing on Extreme Programming (XP) and developing our project through continuous coding, testing and integration.

# **System Description**

**Business Value:**

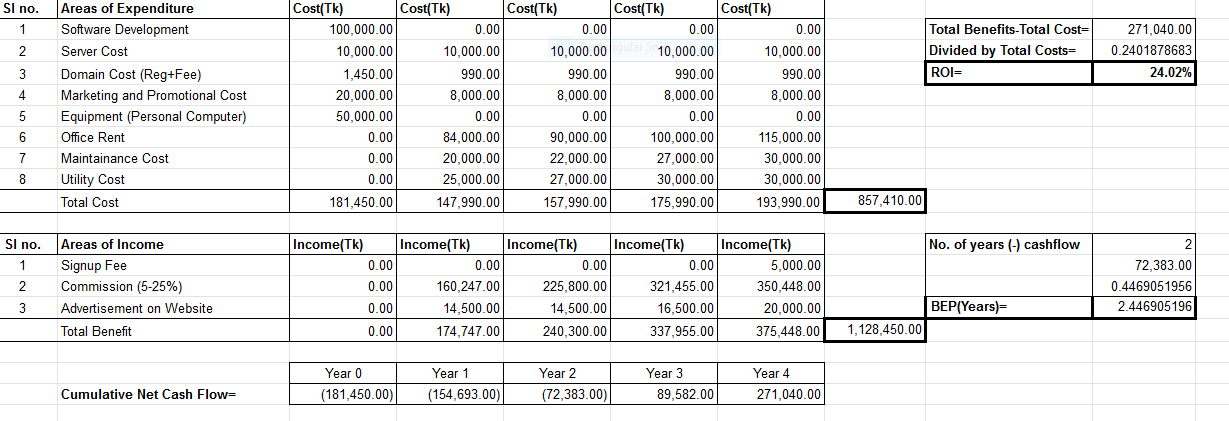
* **Tangible value:**
* We will provide home accommodation service in return of receiving commissions.
* We will receive 15%-25% of the rental fee of first month from the users for using the platform which is made by us where the owners publish their post.
* For the users with special issues we will receive more rental fee.
* **Intangible value:**
* Our platform will reduce valuable time for the tenant to find suitable accommodation anytime anywhere
* It will also reduce time for the tenure to find suitable tenant to share their accommodation.
* Our platform will also reduce hassling for both tenant and tenure.
* For users with special issues our platform will be an effective and easier way.

**Special issues and constraints:**

* Tenants under categories like smokers, alcohol drinkers etc will be strictly maintained.
* Tenants with special medical issues will be handled separately.
* Female tenants will also be maintained separately.

**Feasibility Analysis:**

**Return on Investment and Break Even Point:**



**Software Cost Estimation:**

**System size estimation:**

**System Components (Functional points):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Description |  | Complexity | | |  |
| Total Number | Low | Medium | High | Total |
| Inputs | 31 | 18 x 3 | 7 x 4 | 6 x 6 | 118 |
| Outputs | 16 | 10 x 4 | 4 x 5 | 2 x 7 | 74 |
| Queries | 12 | 2 x 3 | 6 x 4 | 4 x 6 | 54 |
| Files | 25 | 6 x 7 | 7 x 10 | 12 x 15 | 292 |
| Program Interfaces | **9** | **3 x 5** | **2 x 7** | 4 x 10 | 69 |
| **Total Unadjusted Function Points (TUFP):** | | | | | 607 |

**Overall System:**

|  |  |
| --- | --- |
| Data Communications | 3 |
| Heavy use configuration | 0 |
| Transaction rate | 2 |
| End-user efficiency | 3 |
| Complex processing | 1 |
| Installation ease | 0 |
| Multiple sites | 0 |
| Performance | 3 |
| Distributed functions | 0 |
| Online data entry | 1 |
| Online update | 2 |
| Reusability | 0 |
| Operational ease | 2 |
| Extensibility | 0 |
| **Total Processing Complexity (PC):** | 17 |

Adjusted Project Complexity (APC): 0.65 + (0.01 x 17) = 0.87

Total Adjusted Function Points (TAFP): 0.87 (APC) x 607 (TUFP) = 528

**Lines of Code:**

|  |  |  |
| --- | --- | --- |
| **Language** | **Approximate lines of code per function** | **Total lines of code** |
| HTML | 25 | 13200 |
| CSS | 18 | 9504 |
| php | 20 | 10560 |
| JavaScript | 15 | 7920 |
| SQL | 2 | 48 |
|  | Total: | 41232 lines of code |

Our software system is considered to be developed in semi-detached mode with 41232 lines of code. Therefore, using the Constructive Cost Model (COCOMO) using the values below the effort needed to complete the system and the time and number of persons required to complete the system has been determined below.

The co-efficient of ab, bb, cb, db for the three modes of COCOMO are-

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Software project | ab | bb | cb | db | | Organic | 2.4 | 1.05 | 2.5 | 0.38 | | Semi-detached | 3.0 | 1.12 | 2.5 | 0.35 | | Embedded | 3.6 | 1.20 | 2.5 | 0.32 | |

Estimated Effort= 3.0 \*{41.232 (Kilo Lines of code) }1.05

= 148.97

= 149 person-months of effort (nearly)

* Estimated time required= 2.5 \* (149)0.35

= 14.41 months

= 15 months (nearly)

* Estimated persons required= 149/15

= 9.93 persons

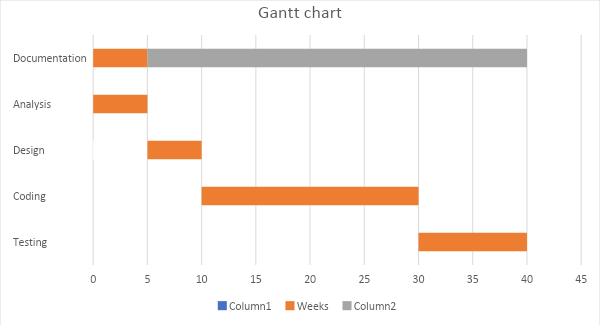
= 10 persons (nearly)

|  |  |  |  |
| --- | --- | --- | --- |
| **TASK** | **DURATION (weeks)** | **PRECEDENCE** | **STAFF REQUIRED** |
| **Analysis** | 5 | **-** | 3 |
| **Design** | 5 | Analysis | 2 |
| **Coding** | 20 | Design | 8 |
| **Testing** | 10 | Coding | 5 |
| **Documentation** | 5 | **-** | 2 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Task** | **Duration** | **ES** | **EF** | **LS** | **LF** | **Slack Time** | **Critical Task** |
| Analysis | 5 | 0 | 5 | 0 | 5 | 0 | **YES** |
| Documentation | 5 | 0 | 5 | 0 | 40 | 35 | **No** |
| Design | 5 | 5 | 10 | 5 | 10 | 0 | **YES** |
| Coding | 20 | 10 | 30 | 10 | 30 | 0 | **YES** |
| Testing | 10 | 30 | 40 | 30 | 40 | 0 | **YES** |

**Staff Utilization factor:** (3\*5 + 2\*5 + 8\*20 + 5\*10 + 2\*5) / (2\* (5 + 5 + 20 + 10))

= 3.06

****

**Timeline:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Analysis** | **Design** | **Coding** | **Testing** | **Documentation** | **Size** | **Effort** | **Productivity** |
| **Front-end Design** | 2 weeks | 2 weeks | 4 weeks | 2 weeks | 2 weeks | 22700 lines | 28 weeks | 3250 lines/month |
| **Back-end Design** | 4 weeks | 2 weeks | 10 weeks | 7 weeks | 2 weeks | 18480 lines | 25 weeks | 2900 lines/month |
| **Database Design** | 2 weeks | 2 weeks | 2 weeks | 3 weeks | 2 weeks | 48 lines | 11 weeks | 16 lines/month |

**Functional and Nonfunctional requirements of the system:**

|  |  |  |
| --- | --- | --- |
| **SL** | **Functional** | **Non-Functional** |
| 1. | 3 User groups, accommodation advertisers, accommodation seekers and Company staff user group. | Secured access with https and encrypted payment solutions. |
| 2. | Company staff user group who can add, delete, update user accounts if they have problems. | The system must be able to work in every web browsers on every platform. |
| 3. | Search by apartment size, accommodation facilities, area and different other options. | The system must be running 24 hours every day in the year with data backups performed during night. |
| 4. | Reservation and payment system for accommodation searchers. | Only admin permission users can see every detailed personal information but no encrypted information. |
| 5. | Accommodation rating system, separate ratings by company staff and users. | The system should support every image format except raw image formats like dng. |
| 6. | Accommodation advertisers can share pictures of their apartment. | The website will be designed using html, javascript, css and will have php handling the mysql database. |

**Risk factors:**

The risk problems that could affect the implementation of this software project has been briefly discussed below:

**Schedule risk:**

Risks that originate when project tasks do not follow schedule in completion and they are not addressed properly.

* Wrong time estimation during calculating the Cost Constructive Model (COCOMO)
* Resources are not tracked properly so the division of tasks are improper
* Failure to identify complex functionalities and time required to develop those functionalities

**Budget Risk:**

May occur mostly in economic feasibility analysis of the software project.

* Wrong budget estimation.
* Cost overruns
* Project scope expansion

**Operational Risks:**

Risks of loss due to improper process implementation failed system or some external events risks.

* failing to address priority conflicts
* failing to resolve the responsibilities
* Insufficient resources
* No communication in the team.

**Technical risks:**  
Technical risks generally leads of this project will result in failure of functionality and performance.

* Continuous changing requirements may come from client
* No advanced technology available or the existing technology is in initial stages
* The product is too complex to implement
* Difficult project modules integration

**Programmatic Risks:**  
These are the external risks that may affect the project beyond the operational limits. These are all uncertain risks and are outside the control of the program.

* Running out of the fund.
* Market development
* Changing customer product strategy and priority
* Government rule changes.

**Conclusion:**

House rental system has been revolutionized by air-bnb which first included the idea of accommodation sharing along with the regular house and apartment rents, but still it faces not competition in its field. Therefore as a competitor with a different approach to providing houses especially in case of bachelors, students, smokers who find difficulty in finding a living place that they can truly call their home. Web based rental system for accommodation has gained a lot of business value where most companies provide only certain features which are gaining business value as determined through lots of online researches done by business analysts and system anlaysts.

**Reference:**

* M. Nusrat, T. Nawshin, C. Amitabha. (2016). Home Rental System Implementing Constraint Satisfaction Problem. Retrieved from

<http://dspace.bracu.ac.bd/xmlui/bitstream/handle/10361/7714/13101075%20%26%2013101070_CSE.pdf?sequence=1&isAllowed=y>

* Jeroen Oskam, Albert Boswijk, (2016). Airbnb: the future of networked hospitality businesses, Journal of Tourism Futures Vol. 2 Issue:pp.22-42, https://doi.org/10.1108/JTF-11-2015-0048

Rental Housing: A much needed housing option for the poor. (2011), ISBN: (Volume): 9/78-92-1-132322-1