1. **Introduction**
   1. **Description of the project**

It is basically a platform where we use to provide each and everything on rent. Users can read and upload posts which contain the information of the rentable item. This platform will provide information to the users regarding wide variety of item on rent nearby. Users will be able to get the details like contact details of the owner of the item. Items include the rooms, houses, cars, bikes, furniture and so on.

* 1. **Novelty in the proposed project work**

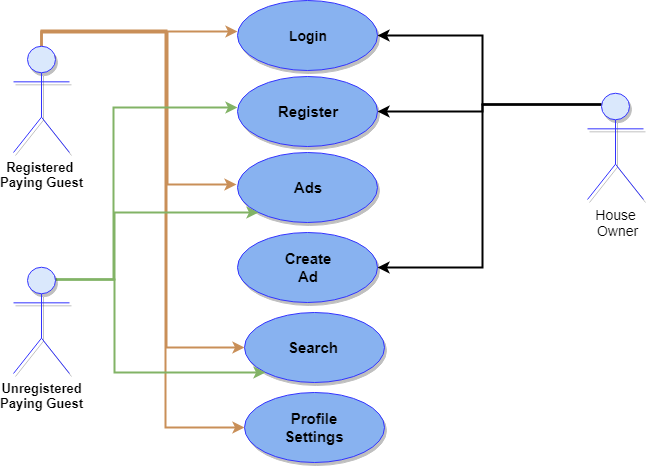
Any app that provides everything on rent still doesn’t exist on play store. There are numerous apps that provide houses and homes on rent. But none of them provides each and everything on rent. We are going to provide each and everything on rent that includes cars, bikes, tents, shops, houses, rooms and so on. Innovative thing about the app is we are going to make a portal which will be specifically known for rent items only.

* 1. **Research gap that the proposed project work intends to fill**

Existing systems or portals are failed to take care of small things that could be provided on the rent. OLX, Quickr.com these are the apps that took initiative to provide rooms and houses on rent but they don’t provide other things like tents, vehicles, clothes etc. on rent. Our portal will try to fill this gap and to provide the users a flexible portal with different options.

Mention the expected project outcomes:

1. Users will get a portal specifically known for rent items.
2. Users can earn money by uploading the post
   1. **Use Case Model**



1. **System Description**
   1. **Customer/User Profiles**
   2. **Customer and User Dependencies**

The application RentPay is the user dependent application. The entire database will be created by the user of the application itself. As the application shows the items available on rent, this is only possible if there will be providers and borrower on the application.

The Application will act as the bridge between the two types of users. The very first type of user of the application is the “Provider”.

Provider: Provider will be providing the items on rent through our application. The application has a great dependency on this type of users as we need more and more of rent providers so that we can have higher variety. Now provider needs to login first into the application for maintaining the authentication of the advertisement. After the user logged in he/she can post the ad for his/her item. They can also search for the items on the application.

The second type of user is the “Borrower”.

Borrower: Borrower is the one who is seeking the desired items. Now the usage of application through borrower is directly dependent upon the type of data that is visible on the application. If there will be variety of items available on the application then it would be more exciting for user. Now the variety and availability is entirely dependent upon the provider’s advertisements. This type of users can skip the login option and can surf, search depending upon their conditions and needs.

The application is entirely dependent upon the user to work. The application is the bridge whose solidity is directly proportional to the user-base.

Once the provider’s and borrower’s conditions match then they can contact each other. After that application plays no role and have no responsibility about the deal.

* 1. **Functional Requirements**
     1. **Create a new account**

**Description:**This function registers a new user to the ‘Rent Pe’ by creating a new account.

**Input:**Details of the user such as name, age, country (location), email-id, and phone-no.

**Output:**Displays message for “Successful Registration” and redirects to the customized home page.

**Process:**It is used for registering new users to the RentPe’s user-details database and on successful registration, logs in to their account and redirects to the feed page.

* + 1. **Signup using any third-party account**

**Description**: This function is used for signing up new users to RentPe using third-party accounts such as Gmail and Facebook.

**Input**: Email-id and Password

**Output:** Displays message for “Successful Registration” and redirects to the customized home page.

**Processing:** It directs the user to third-party’s login page for log in process and makes sure that user accepts the terms and conditions. Retrieves information from the third-party and creates a new account using those details and confirms those from the user along with getting any missing details.

* + 1. **Login**

**Description:** This function is used for logging into the RentPe account.

**Input**: User-id and Password

**Output:** Displays tailored home page on successful log in.

**Processing:** If the user-id and password entered by the user is correct then, log in the user and display the customized home page. Otherwise, display the error message and remain on the login page(default).

* + 1. **Bookmarks**

**Description**: This function is used for bookmarking any post.

**Input:** Unique post-id (No user input required).

**Output:** Displays the bookmarked symbol within the context of the post.

**Processing:** It allows the user to bookmark any post which was posted by any specific person, which can be viewed at any point of time, unless it exists in the system.

* + 1. **Ratings**

**Description:** This function is used for giving ratings to any user’s post.

**Input:** Index in the range 0-5, both inclusive.

**Output**: Displays the updated overall rating statistics.

**Processing:** It is used for giving ratings to any post. User must has full access to the rating given by him/her, allowing to update it any time and the same must be reflected.

* + 1. **Edit**

**Description:** This function is used for editing any of the previously posted ad-post by the user.

**Input:** Updated content as a string.

**Output:** Displays the updated content along with last edited time.

**Processing:** It allows the user to edit and update any of the previously posted ad-post and displays the updated content along with the last edited time.

* + 1. **Comments**

**Description:** This function allows the user to make comments/reviews to any other user’s posts.

**Input:** Comment as a string.

**Output:** Posts the user’s comment to the concerned user and user’s post.

**Processing**: It allows the user to make comments/reviews to any other user’s post, to enhance the content’s description or clarity.

* + 1. **Notifications**

**Description:** This function is used for displaying the notifications to the user.

**Input:** User-id used for logging in the account (No user input required).

**Output:** Displays the notification page for the user.

**Processing:** It displays all the user notifications that comprises of ratings, comments, views, to the user’s ad-posts, etc. made by the other users.

* + 1. **Settings**

**Description:** This function is used for changing the user’s account preferences.

**Input:** User-id used for logging in the account (No user input required).

**Output:** Displays the settings page.

**Processing**: It is used for changing any of the user’s account preferences.

* + 1. **Activity**

**Description:** This function is used for displaying the user’s activity.

**Input:** User-id used for logging in the account (No user input required).

**Output:** Displays the activity page.

**Processing:** It displays the activity of the user that comprises of user’s posts, ratings, edits, comments, etc. in the user’s timeline

* + 1. **Search**

**Description:** This function is used for searching and displaying the results.

**Input:** Keywords (String) to be searched.

**Output:** Displays the search results after applying the filters.

**Processing:** It searches for the given keywords and displays the relevant information after applying the filters.

* + 1. **Filters**

**Description:** This function is used for applying any combination of filters on the search results.

**Input:** User needs to apply any filter and an array of posts on which it is to be applied.

**Output:** Displays the results after applying the specified filters.

**Processing:** It displays the search results after applying the specified filters and if no filter is specified then by default sort in descending order of the geographical distances from the user.

* + 1. **Post a new Ad**

**Description:** This function is used for posting a new Ad.

**Input:** Form having all requires fields to be filled such as title, pics, location, price, etc.

**Output:** Displays the new post in formatted manner.

**Processing**: It takes the post details and generates a unique post-id and posts the ad-post to the database servers and displays it in the formatted manner.

* + 1. **Help**

**Description:** This function is used for displaying the help contents related to the topic.

**Input:** Keywords to be searched (String)

**Output:** Displays the help content regarding the entered keywords.

**Processing:** It takes keywords as an input and looks for the help contents regarding the input and if found displays it, else displays the error message, accordingly.

* + 1. **Feedback**

**Description:** This function is used for giving feedbacks to the RentPe’s administrators/developers by the users.

**Input:** Feedback form containing information regarding subject and it’s description.

**Output**: Displays thanks message and takes back the user to the home page.

**Processing:** It takes the feedback form from the user and submits it to the admins/developers.

* + 1. **Forgot Password**

**Description:** This function provides user to recover his/her forgotten password.

**Input:** User-id or email-id and mobile number.

**Output:** Details confirmation message.

**Processing:** It gets input from the user as user-id or email-id, and mobile number. Checks for the correctness and sends the OTP(One Time Password) to the registered email-id and mobile number along with access to change password using the provided OTP.

* + 1. **Logout**

**Description:** This function is used for logging out of the RentPe account.

**Input:** User-id used for logging in the account (No user input required).

**Output:** Displays home page, as if viewed by any guest.

**Processing**: Logs out the user from the account and displays the home page, as if the user was unknown (guest) to the system.

* 1. **Non-Functional Requirements**



### Performance Requirements

### System shall be available from all over the world by all the times, only restricted by the down time of the server on which the system runs. Being a social networking of advertisements, any interruption in the sharing chain will cause people to give up on Facebook, therefore it is essential that the system shall be available by all the times. System shall not be affected from the number of active users in the system until half of the registered users become active. Being a worldwide network, assuming, that half of the registered users are reaching to the website is a legitimate and necessary requirement.

### In case of a of a hardware failure or database corruption, a replacement page must be shown. Also in case of a hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the administrator.

### Design Requirements

### Design of the system shall arrange the content size as compatible for different platforms, such as mobile phones, tablets of different specifications and the versions of the underlying software architecture. Design of the system shall let high level of mobile access.

### [Optional]: Design of the system may let different languages to be shown without affecting the general layout and operations. Being a worldwide network, different language sets shall be able to shown as the main language of the application without creating any obstacles on the operations.

### Security Requirements

### As the account of each user is protected by a user id and password, any unauthenticated access to the account must not be allowed. Privacy policies are also applicable to each and every user.

### The system uses SSL (secured socket layer) in all transactions that include any confidential customer information.

### The system must automatically log out customers after a period of inactivity, for security purposes.

### The system should not leave any cookies/caches on the customer’s computing device containing the user’s id, password, etc. in plain text form.

### The system’s back-end servers shall only be accessible to authenticated administrators.

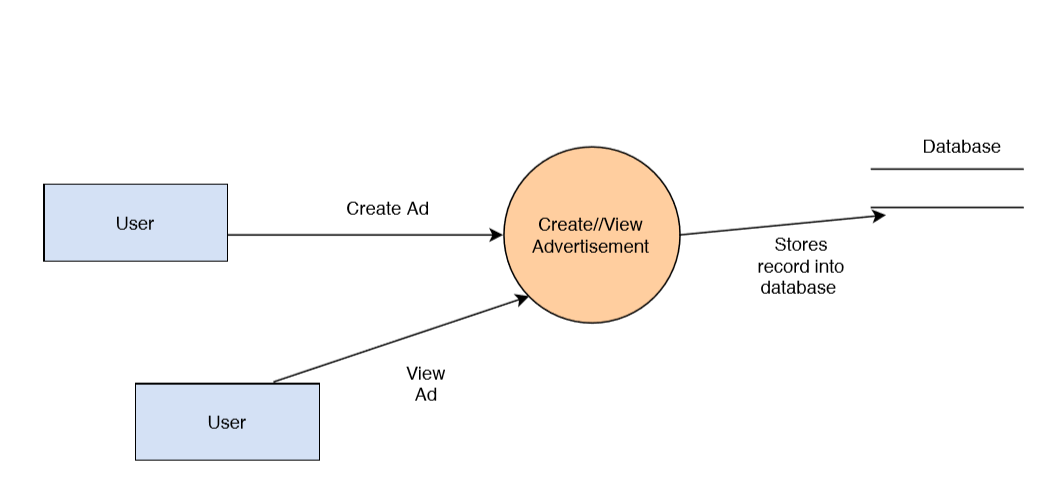
### Reliability

### The system provides storage of all databases on redundant computers with automatic switchover. The reliability of the overall program depends on the reliability of the separate components. The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes.

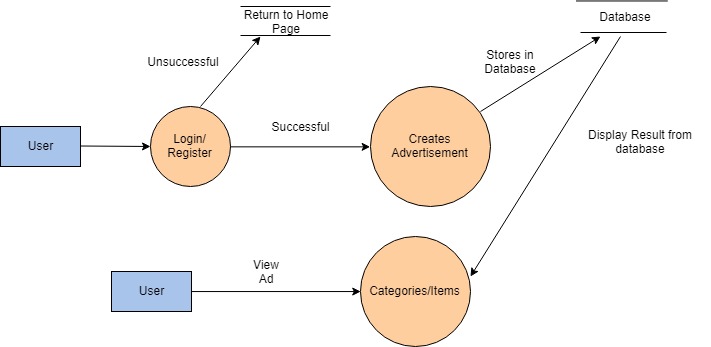
### Maintainability

### A commercial database is used for maintaining the database and the application server takes care of the site. In case of a failure, a re-initialization of the program will be done. Also the software design is being done with modularity in mind so that maintainability can be done efficiently.

1. **Design**
   1. **System design**
      1. **E-R diagram**
      2. **DFD’s**

**Level 0:**

**Level 1:**



1. **Scheduling and Estimates**

