**USE CASE DIAGRAM**

Room Requests

Occupancy Map

File Storage

Students

Administrators

Database

Figure : Room Storage Use Case

|  |  |
| --- | --- |
| Use Case Name | Requesting a new room |
| Participating Actors | Students, Administrators |
| Flow of Events | 1. Student logs into Housing website. 2. Student navigates to request form 3. Request form is filled out. 4. Form is sent to the database. 5. Administrators are alerted to new request by e-mail 6. Admins open program 7. System displays list of unanswered room requests 8. Admin selects a room request 9. System displays information pertaining to that request 10. Admin accepts or rejects request 11. Letter sent to student alerting them of response. 12. (If accepted) System replaces student’s current file with new one 13. Student’s file removed from list of unanswered room requests 14. List of unanswered room requests displayed |
| Alternate Flows | 3a. User cancels request.  3b. User is returned to the Housing homepage.  11a. If rejected, rejection e-mail sent to student |
| Entry Condition(s) | Student has a valid CLID and password. |
| Exit Condition(s) | Form is correctly filled out and entered; alternately, user cancels request. |
| Special Requirements | N/A |

|  |  |
| --- | --- |
| Use Case Name | View occupancy map |
| Participating Actors | Administrators |
| Flow of Events | 1. Administrator accesses program 2. Select building from a drop down list 3. Select floor from another drop down list 4. Search for floor 5. System displays map |
| Alternate Flows | 2a. Enter name of building through type  3a. Enter floor through text  5a. System displays an error message if floor terms not valid  5b. Display search page again |
| Entry Condition(s) | Chosen floor plan is valid and the database has been updated with most recent information. |
| Exit Condition(s) | User exits the program. |
| Special Requirements | N/A |

|  |  |
| --- | --- |
| Use Case Name | Store hard copy files into database |
| Participating Actors | Administrators |
| Flow of Events | 1. Administrator opens program 2. System displays options 3. Administrator selects option to add physical to database 4. System waits for file to be scanned 5. Admin scans file into the program 6. Program converts file into electronic format 7. Admin names file 8. File is uploaded to database and categorized |
| Alternate Flows | N/A |
| Entry Condition(s) | Physical file is correctly formatted and filled out |
| Exit Condition(s) | A file is successfully entered or the process is cancelled |
| Special Requirements | Requires a scanner |

**RATIONALE**

It was determined that, in order to efficiently perform any of the client’s tasks, a database would be necessary. Since all of the tasks needed one, it was decided that all of the projects would be completely based around the database. The only extra things required would be interfaces unique to the individual projects.

**NON-FUNCTIONAL REQUIREMENTS**

**Product requirement**

The execution speed, reliability and accessibility of the system are described here.

The room-change system should be available to all current residents on campus during weekdays and weekends. Downtime for site maintenance shall not exceed 24 hours.

The system should be easy to use by residents and should be organized in such a way that user errors are minimized.

**Organizational requirement**

Process standards used by the system are described here.

Users of the room-change system shall authenticate themselves using their CLID and user-defined password.

Priority of the room-change service should follow the first-come, first-served practice.

**External requirement**

Interoperability and legislative requirements are described here.

The system shall implement resident privacy provisions.

Maintenance should be done on the system regularly.

**EVIDENCE OF CONFIGURATION MANAGEMENT**

Git Repository: https://github.com/brandinjefferson/CMPS453-Docs



**REFERENCES**

Badgerati. “Software Engineering – Use Case Diagrams / Descriptions.” *Computer Science Source*. Computer Science Source, 22 Nov 2009. Web. 29 Sep 2014.