|  |
| --- |
| CMPS453 – University of Louisiana at Lafayette |
| Design |
| UL Housing Project |

|  |
| --- |
| Brandin Jefferson, Issa Samake, Zach Danjean, Yee Wong, Jaquincy Nelson, Brian Okoye  10-23-2014 |

# ABSTRACT

The document below describes both the visual and back-end aspects of the project that prior documents have not broached. The features that are being described; each separate diagram that shows the processed work in both static and dynamic manners. A prototype of the actual user interface is also provided within the document.

Table of Contents

[ABSTRACT 1](#_Toc404632278)

[LIST OF FIGURES 2](#_Toc404632279)

[INTRODUCTION 3](#_Toc404632280)

[GUI (Graphical User Interface) DESIGN 4](#_Toc404632281)

[STATIC MODEL SEQUENCE DIAGRAMS 6](#_Toc404632282)

[DYNAMIC MODEL SEQUENCE DIAGRAMS 7](#_Toc404632283)

[RATIONALE FOR DETAILED DESIGN MODEL 10](#_Toc404632284)

[TRACEABILITY FROM REQUIREMENTS TO DETAILED DESIGN MODEL 10](#_Toc404632285)

[REFERENCES 10](#_Toc404632286)

# LIST OF FIGURES

[Figure 1 - GUI: Login Screen 4](#_Toc404952219)

[Figure 2 - Student Interface 5](#_Toc404952220)

[Figure 3 - Admin Interface 5](#_Toc404952221)

[Figure 4 - Static Diagram: Room Change (Student)/(Admin) 6](#_Toc404952222)

[Figure 5- Dynamic Diagram: Create New ID 7](#_Toc404952223)

[Figure 6- Dynamic Diagram: Request New Room 8](#_Toc404952224)

[Figure 7- Dynamic Diagram: View Floor Map **Error! Bookmark not defined.**](#_Toc404952225)

[Figure 8- Dynamic Diagram: Decide Room Requests 9](#_Toc404952226)

# INTRODUCTION

The purpose of the detailed design document is to provide a low level description of the change room system, providing insight into the structure and design of each component. This document is meant to equip the reader with a solid understanding of the inner workings of the room change system.

* 1. Goals and Objectives

The purpose of the room change system is to facilitate the process of requesting room change for both the occupants and the housing employee in charge. Thus, the occupant can request a room change on the webpage and his/her request will be send to a queue where it will be review by the UL housing employees based on the requested date and the availability of the rooms. The system will also allow the housing worker to update he database by adding, removing and switching occupants. Finally, it will keep track of room availability.

* 1. Projects overview and scope

The room change system will be composed of 2 main components: the database and the interface. The system will have 2 user interfaces, one for the occupants to request room change and one for the UL housing workers to access and modify the databases. The system will also have a 4 major databases: one for the occupants’ information, one for the queue of requests, one for the administrators and one for the building and rooms.

The detail design document will cover 3 primary parts. These include the GUI Design, the static model diagrams, and the dynamic model diagrams. In addition to these are explanations for the designs and an explanation of how the models connect with the original architecture.

# GUI (Graphical User Interface) DESIGN

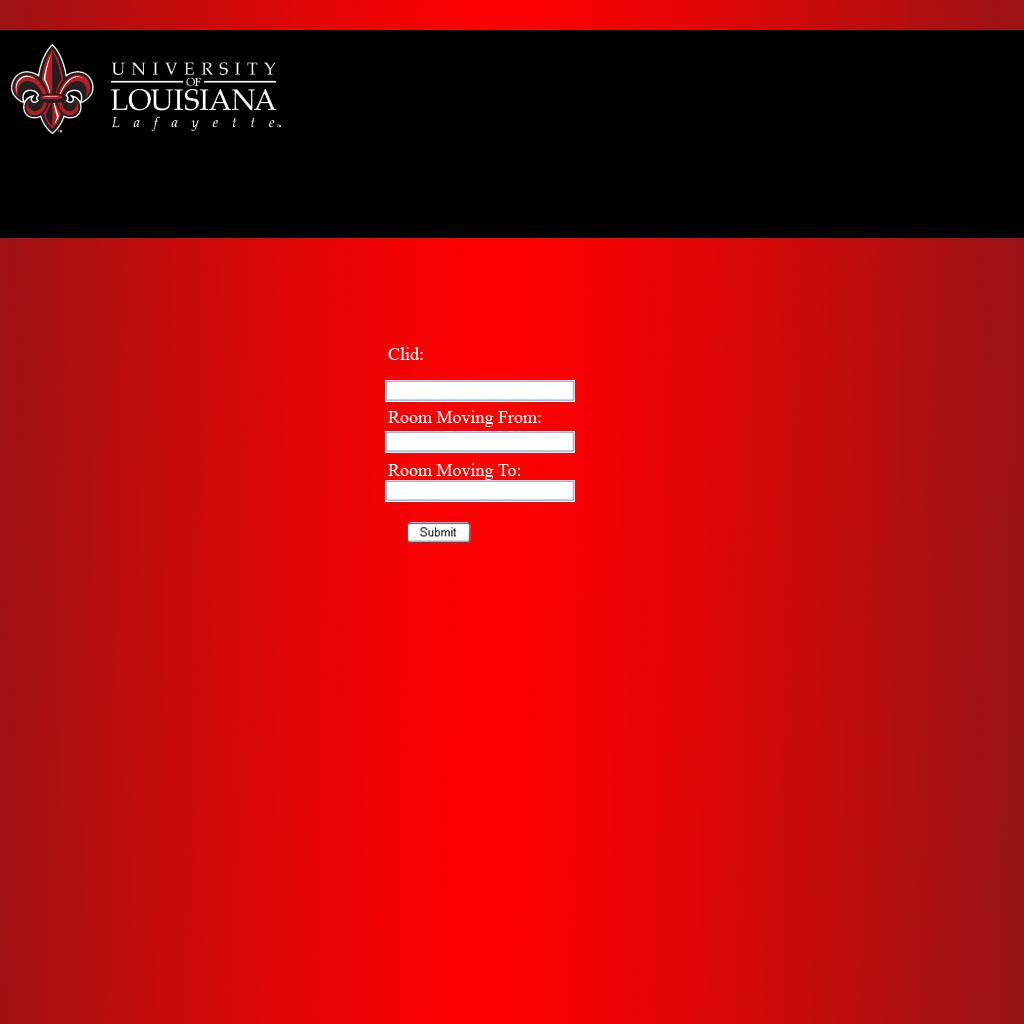


Figure - GUI: Login Screen

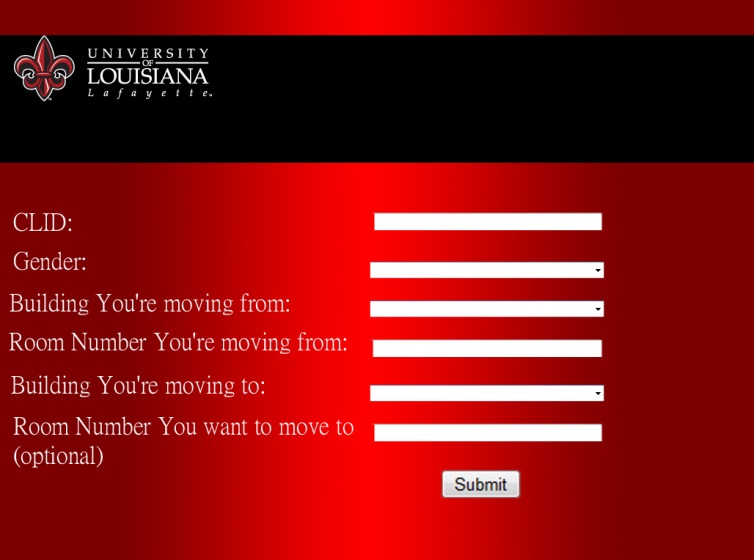


Figure - Student Interface

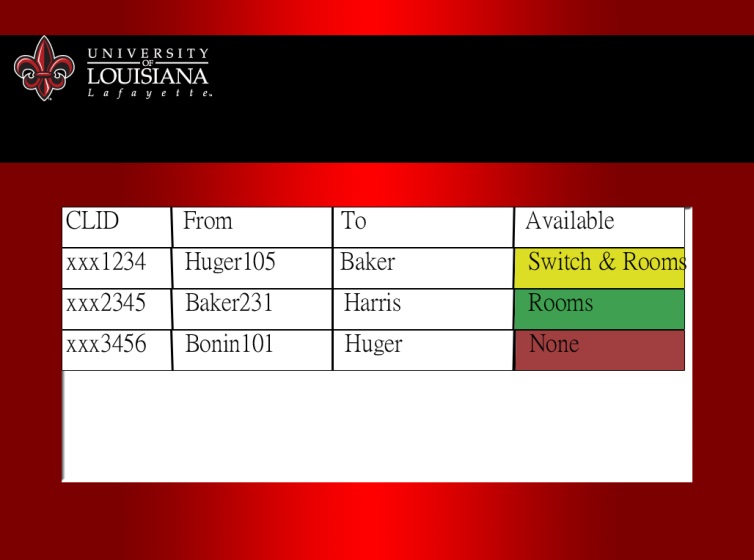


Figure - Admin Interface

# STATIC MODEL SEQUENCE DIAGRAMS

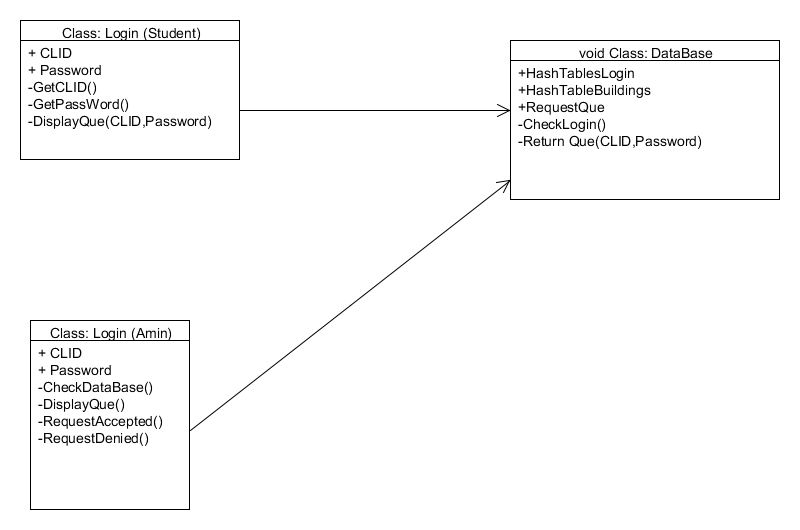


Figure - Static Diagram: Room Change (Student)/(Admin)

# DYNAMIC MODEL SEQUENCE DIAGRAMS

sd Create New ID (void) : void

recordlist : StudentRecord

ids : StudentID

verifyID (id, password)

: boolean

*exists*

alt

[exists == false]

Login (id, password)

: void

mail : MailVerifier

createID(email, password)

: void

verifymail(email):

boolean

*valid*

alt

[valid == false]

notvalid() : void

errorMsg()

: void

else

AddID(): void

else

displayPage() :

void

Figure - Dynamic Diagram: Create New ID

Verifies a student’s identity if their CLID has not been used before by sending an email to their school email address. If it’s found to be valid, then the student is able to continue.

sd Request Room Change (void) : void

holder : Interface

db : HousingRecord

selectRoomChange

displayOptions

available(building, room)

*available*

alt

[available == true]

updateAvailability(id)

successMsg

exit

[else]

updateAvailability(id)

notFirstMsg

exit

rq : Request Queue

add

Figure - Dynamic Diagram: Request New Room

When a student registers for a room or building, the interface will send a query to the database to determine the availability of the room or any rooms respectively. A message displaying whether there is a chance at the moment or not is then printed to the screen.

sd Decide Room Requests

Interface Manager

DisplayRequests

Request Queue

*Request Queue*

DecideRequest

removeRequest

loop

sendMail

Figure - Dynamic Diagram: Decide Room Requests

After the admin logs in, they will have a view of every confirmed request. From there, they can accept or deny a request, the result of which will be an email being sent to the student with the correct response.

# RATIONALE FOR DETAILED DESIGN MODEL

We decided upon the given design by viewing other websites and examples. The final product that we’re creating is not unique by any means and, as such, already has plenty of use. The group chose the aspects that applied most appropriately, which placed in their simplest form. The ideology of “the simpler, the better” was heavily in play here due to the potential massive amount of traffic the client may receive in comparison to the number of employees available. So a minimum number of tasks are available at any one time for a user to choose, whether they be a student or administrator in order to streamline the process of whatever needs to be accomplished.

# TRACEABILITY FROM REQUIREMENTS TO DETAILED DESIGN MODEL

The requirements document lists all of the features described by sequence diagrams and the user interface in this document. Each feature, save those that are no longer going to be used due to changes in the project parameters, has been implemented in greater detail than was described by the requirements. This was thanks to the group now having a clear set of tools and designs available for use, rather than just a vague idea of how the project should be handled.

# REFERENCES

Bell, Donald. *UML basics: The sequence diagram*. IBM Corporation, 14 Feb. 2004. Web. 26 Oct 2014.