STUDENT RESIDENCE MANAGEMENT SYSTEM

by

Iraba Marie Louise

A thesis submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science (Honours) in Computer Science

University of the Western Cape

2009

Date: June 4, 2009
University of Western Cape

Abstract

STUDENT RESIDENCE MANAGEMENT SYSTEM

by Iraba Marie Louise

Supervisor Committee: Professor H. Nyongesa and Professor I. M. Venter
Department of Computer Science

The University of the Western Cape (UWC) houses approximately 3500 students. A total of twelve University residences, with eight located on the Campus and four off campus. Occasionally, the University would house students off campus if there is a demand for accommodation. The Student Residence Management System is developed to facilitate students to apply for accommodation online and to help the staff to manage the different residence activities such as controlling booking, payments and room allocation. The Student Residence Management System will be able to notify and confirm all room allocations. Room allocation confirmations would be sent by email to students who were given accommodation. The development of this system also focuses on security of the information and privileges and access rights are attributed to students and administrators.
TABLE OF CONTENTS

Table of Contents i
List of Tables iv
List of figures v
Acknowledgments vi
Glossary vii

Chapter 1 1
Sketching the background 1
1.1 Introduction 1
1.2 User characteristics 1
1.3 Current system 2
1.4 Motivation 3

Chapter 2 4
User Requirement DOCUMENT 4
2.1 Introduction 4
2.2 Data collection technique 4
2.3 Problem domain 4
2.4 Users’ problems on the current system 5
2.4.1 Students’ problems 5
2.4.2 Residence Administrators Problems 6
2.5 User expectations from a new computerized system 6
2.6 Limitations to be encountered 7
2.6 Conclusion 8

Chapter 3 9
Requirement Analysis document 9
3.1 Introduction 9
3.2 Designer’s interpretation of the user’s requirements 9
3.3 High-level Constituents Parts 9
3.3.1 Database management 9
3.3.2 Software management 10
3.4 Existing solutions 10
3.5 Alternative Technical Solutions 10
3.6 Best Solution 10
3.7 The Model to be used 11
3.8 Technologies to be used 11
3.9 System testing 12
3.10 Conclusion 13
# Chapter 4
User Interface Specification

4.1 Introduction

4.2 Description of the complete user interface

4.2.1 Login page

4.2.2 Accommodation Application Page

4.2.3 Feedback Page

4.2.4 UWC Admin Page:

4.2.5 ResManager Page:

4.3 How the User Interface behaves

4.4 How the user interacts with the system

4.4.1 How the student interacts with the system

4.4.2 How the Administrator interacts with the system

4.4.3 How the ResManager interacts with the system

4.5 Conclusion

# Chapter 5
Object Oriented Analysis (OOA) or High Level Design

5.1 Introduction

5.2 Data Dictionary

5.2.1 Student Object

5.2.2 Administrator Object

5.2.3 ResAdmin Object

5.2.4 Residence Object

5.2.5 Room Object

5.2.6 PAYMENT OBJECT

5.2.7 BOOKING OBJECT

5.3 Class Diagram

5.4 Conclusion

# Chapter 6
Object Oriented Design (OOD) or Lower Level Design

6.1 Introduction

6.2 Inner Details of Class Attributes and Methods

6.3 State Diagram

6.4 Sequence Diagram

6.5 Pseudo code

6.5.1 display Login Page

6.5.2 Student application

6.5.3 Payment

6.5.4 Allocate

6.3 Conclusion

# References

# Appendices

# bibliography
<table>
<thead>
<tr>
<th>Number</th>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Student Object</td>
<td>28</td>
</tr>
<tr>
<td>2</td>
<td>Administrator Object</td>
<td>28</td>
</tr>
<tr>
<td>3</td>
<td>ResAdmin Object</td>
<td>29</td>
</tr>
<tr>
<td>4</td>
<td>Residence Object</td>
<td>29</td>
</tr>
<tr>
<td>5</td>
<td>Room Object</td>
<td>30</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Number</th>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1:</td>
<td>Login Page</td>
<td>15</td>
</tr>
<tr>
<td>Figure 2:</td>
<td>Accommodation Page</td>
<td>16</td>
</tr>
<tr>
<td>Figure 3:</td>
<td>Feedback page</td>
<td>17</td>
</tr>
<tr>
<td>Figure 4:</td>
<td>UWC Administrator page</td>
<td>18</td>
</tr>
<tr>
<td>Figure 5:</td>
<td>Residence administrator</td>
<td>18</td>
</tr>
<tr>
<td>Figure 6:</td>
<td>Use case diagram</td>
<td>19</td>
</tr>
<tr>
<td>Figure 7:</td>
<td>Student activity diagram</td>
<td>21</td>
</tr>
<tr>
<td>Figure 8:</td>
<td>Admin activity diagram</td>
<td>23</td>
</tr>
<tr>
<td>Figure 9:</td>
<td>ResAdmin activity diagram</td>
<td>25</td>
</tr>
<tr>
<td>Figure 10:</td>
<td>Class diagram</td>
<td>32</td>
</tr>
<tr>
<td>Figure 11:</td>
<td>State diagram</td>
<td>35</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

The author wishes to [Click and type acknowledgments]
<table>
<thead>
<tr>
<th><strong>HTML</strong></th>
<th>Hyper Text Markup Language</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HTTP</strong></td>
<td>Hypertext Transfer Protocol.</td>
</tr>
<tr>
<td><strong>MYSQL</strong></td>
<td>My Structured Query Language.</td>
</tr>
<tr>
<td><strong>OOA</strong></td>
<td>Object-Oriented Analysis</td>
</tr>
<tr>
<td><strong>OOD</strong></td>
<td>Object-Oriented Design</td>
</tr>
<tr>
<td><strong>PHP</strong></td>
<td>Hypertext Preprocessor</td>
</tr>
<tr>
<td><strong>UML</strong></td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td><strong>UWC</strong></td>
<td>University of the Western Cape</td>
</tr>
<tr>
<td><strong>WWW</strong></td>
<td>Word Wide Web.</td>
</tr>
</tbody>
</table>
Chapter 1

SKETCHING THE BACKGROUND

1.1 Introduction
The University of the Western Cape has decided that by 2010, the entire student registration process will be online. This vision implies that the student accommodation system should also be computerized and possibly become online based.

The current accommodation application process involves students to fill a paper-based application form which takes a long time to process. The Student Residence Management System (SRMS) is introduced as a computerized system to ease the pressure on students and administrators. The new residence application system will help to eliminate the paper-based applications thereby speeding up processing as details will be captured in the database. This will help to eliminate the delays caused by the sequential checking of each residence for the availability of rooms. The system will check automatically for the available rooms to help accelerate the room allocation process thus making it painless for the managers to allocate rooms to students.

At the present time, students must check for the successful or unsuccessful list of accommodation application at the residence administrator. Since the new system to be adopted is computerized, it will automatically send notifications to successful and unsuccessful applicants through email.

1.2 User characteristics
Students’ residence management system currently involves three types of users; these are Students, Residence Administrator, and UWC Administrator.
• Student Users: this type of user includes all the UWC students who apply for accommodation. These users will also adopt the online system when applying for their accommodations.

• Residence Administrator: these are members of staff who are responsible of residence management. Their task will be allocating rooms to students and notifying students of their application results by email.

• UWC Administrator: this will be a member of administration of the college who will be in charge of application fees for residence.

1.3 Current system

The current system of residence application is not yet computerized and this has made the process of application to be very slow and tiresome. During application, forms are provided to students who fill them by hand and deposit them to the Residence Office. The staff in charge classifies all the application forms and processes each case at time.

The current system being manual, the management does not have updated records of vacant rooms or occupied rooms. To know whether there are still some vacant rooms, the person in charge has to go to the residence building and check physically. Also the lack of updated information about the room availability and their types causes some students to be allocated to the type of rooms they did not want.

The manual system of residence management has another constraint of not accessing information from the students’ database. The residence management staffs do not get timely information about students who have been suspended from the college, who have failed, students who have not completed registration and payment, unless they go and request this information from the college administration.
1.4 Motivation

The idea of computerizing this system was introduced with a view to address the problems associated with the current residence management system.

This project introduces a new online-based system of application for student residence which will facilitate students to book for the room they want without spending much time on the queue. Also this system will help the management staff to process the application and notify the students in a short time. This must be achieved by enabling the residence management staff to access, timely, all student information from the UWC students database; and this system also will provide a facility of record keeping about room allocations and room availability.
2.1 Introduction
In the previous chapter, the current system and the users of the system were discussed. This chapter tackles the users’ views about the current system and what they expect from the new system. Also, the data collection tools used are described.

2.2 Data collection technique
The data collection instruments used to gather accurate information about the existing system and the requirements for the new system were interviews with stakeholders, questionnaires and observations.

Interviews and questionnaires were administered to students and residence management staff to collect user requirements. Observation of the current existing system was done at the Residence Administration Office in order to find out how the existing system functions, the problems encountered and how they can be solved by the new computerized system.

2.3 Problem domain
The problem domain of this project is the computerization of a university student’s residence management system with a facility of online access by the users.
2.4 Users’ problems on the current system

The stakeholders are students and residence administrators and they encounter different problems with the current system. The information collected from both users is as follow:

2.3.1 Students’ problems

- Currently students register for accommodation at residence office only. In case students are in distant areas, where they can not reach the university, registration for accommodation is quite difficult.

- Paper based application requires students to print, complete and submit the application form at the residence administration. It is difficult because it takes more than three days to register since there is always a long queue.

- Students apply for accommodation and when they are accepted to study at the University, they are told to re-apply even though they applied before and this confuses students and take much time.

- When the students apply for rooms, there is confusion because some students are given the type of the rooms which they did not apply for. This is because administration does not always have updated records of vacant rooms and their types.

- There is no electronic complaint or suggestion box (anonymous).
The students can not make complaints, suggestions or comments about the problem they face with the system and also they cannot send email to the administrator automatically.

2.3.2 **Residence Administrators Problems**

- Allocation of same room to two different people.
  Users make a mistake of giving a single room to different students at the same time. When student makes booking, the system doesn’t show if the room is vacant or not.

- Tedious capturing data and processing of the paper-based student applications.
  When applying into the residence database, some time the users are not able to read the handwriting of the students and that cause them to enter the wrong information into the system.

- Sometime the users make mistakes when tracing students who paid half or the full amount.

- Users do not have access to database to get the information about students eg. those who fail, those who leave the college for different reasons etc.
  When the residence administrator wants the information about students they go to the faculty office and administration.

2.4 **User expectations from a new computerized system**

According to the investigations conducted, the following are students’ expectations from the new system:
• The interface must be simple, easy to understand and use.
• The system must be secured
• Remote access to the system of residence application.
• The system must be able to inform the students whether the application has been successful or not. Through email students should be able to receive their room numbers.

According to the survey, the administrators want the system to have the following features:

• The system must be secured
• The system must identify the occupied rooms as well as unoccupied ones.
• The system must be integrated/linked to the college students’ database for easy retrieval of student’s information (name, address, id number etc), academic, financial records and application/admission status.

2.5 Limitations to be encountered

During system implementation, the designer will not have direct access to the UWC students’ database for easy retrieval of student’s information (name, address, id number, academic and financial records and application/admission status) because of the university policy. However, a similar database will be created as to help the system developer to test for the new system.
2.6 Conclusion

The student residence management system will be the best solution to residence management problems as it will provide easy online application, so that students can apply from home or anywhere else via internet. The online system will also provide quick and reliable registration process hence reducing the load of work done by the staff and saving time for students who apply.
Chapter 3

REQUIREMENT ANALYSIS DOCUMENT

3.1 Introduction

In the previous chapter the user requirements were identified. In this section we will focus on the system and software requirements needed for implementation of the new system. Also the software development tools which will help to address the problems will be identified.

3.2 Designer's interpretation of the user's requirements

- Students want remote access to the system of residence application. This means that the system has to be web-based to enable them to apply from home.
- The system will need a database because it involves a lot of information which has to be stored and retrieved later, for example information about types of room and their availability.

3.3 High-level Constituents Parts

3.3.1 DATABASE MANAGEMENT

The database will be managed by the administrator and it will have the following characteristics:
- The Database shall be accessible by the software.
- The Database shall allow users to store and search for data
- The Database shall allow users to modify stored data.
- The Database shall allow users to delete stored data.

### 3.3.2 Software Management

- The software shall be accessible from all platforms.
- The software shall be able to interact with the database to retrieve and modify data.
- The software shall be able to add data to the database.
- The software shall be able to edit data in the database.
- The software shall be able to delete data from the database.
- The software shall be able to retrieve data from the database.
- The software shall be able to create reports based on information in the database.

### 3.4 Existing solutions

Currently the University has a website where students find and print application form. Some members of administration use the database of UWC where information about students is found. This information is taken into consideration while processing the applications made by students.

### 3.5 Alternative Technical Solutions

The whole system of application for residence should be computerized and preferably must be a web-based system.

### 3.6 Best Solution

The Online-based Student Residence Management System will be easy flexibility to users, consistent and also it will fulfill the user requirements.
3.7 The Model to be used
In order to analyze the user requirements, UML will be used. UML will help to model the exact activities of the new system, to help understand very well how the system will work and to present a list of tasks to the user. Furthermore, UML will help to break down the project to make it much easier to develop.

3.8 Technologies to be used
It is recommended to use a Web-based technology for the system (using PHP & MySQL with Apache as the web server). The advantage is that the system developed using this technology is:

- easy to use (with user-friendly interfaces)
- free (doesn’t require any license)
- cheaper
- easy to manage and maintain

- **PHP**
The PHP Hypertext Preprocessor allows web developers to create dynamic content that interacts with databases. (http://www.php.net)
It’s a server-side scripting language.

- **MySQL**
MYSQL is an open source relational database management system. It is based on the structure query language (http://www.mysql.com), it is consistent fast performance, high reliability and ease of use.

- **Apache**
The apache is a freely available Web server that is distributed under an "open source" license.
• **PHPMyAdmin**
As the user graphic, it is the interface free and convivially realized in program language (PHP) and easy to manage the MySQL database on the server (www.phpmyadmin.net)

• **JavaScript**
Is a scripting language used to enable programmatic access to objects within other applications. It is primarily used in the form of client-side JavaScript for the development of dynamic website. (www.wikipedia.org)

• **Hypertext Transfer Protocol (HTTP)**
Is an application-level protocol for distributed, collaborative, hypermedia information system. Its use for retrieving inter-linked resources led to the establishment of the World Wide Web. (www.wikipedia.org)

### 3.9 System testing

• **Paper Prototyping**
Drawing of the system will be put on paper and shown to the users so that they can test the system. This will be useful because;
  - Paper Prototyping saves time and money because it allows developers to test the system before code is written.
  - Allows easy and cheap amendment to the design.
  - Lowers support burden and increase an overall quality of the software.

• **Usability Testing**
Usability testing is a technique used to evaluate a product by testing it on users. During usability testing random users will be brought in to test the system.

- **Complete System Testing**

  Black box testing method will be used to test the complete system. Appropriate data will be selected as per functionality and test it against the functional specifications in order to check for normal and abnormal behavior of the system.

### 3.10 Conclusion

After examination of possible solutions, we chose the web-based system as it will be able to address problems encountered by users within the current residence administration system. The proposed system is to find optimal matching for user requirements, and management of residences. System implementation will be carried out using PHP, MySQL, PHPMyAdmin and Apache, which are the open source applications.
Chapter 4

USER INTERFACE SPECIFICATION

4.1 Introduction

From the previous chapter we discussed about the user requirements specification, the proposed solutions and alternatives to those requirements. This chapter describes the user interface, how it works and how the user interacts with the program. Also the images of the user interface pages are included as to show how application looks like.

4.2 Description of the complete user interface

The application user interface has of the following pages:

4.2.1 LOGIN PAGE

The Login page consists of two text boxes, namely Username and Password; and a Login command button allowing the users to log into the system. The login page helps the users to login as a Student who makes the application, as an Administrator who receives application fees for accommodation or as a ResManager whose duty is to allocate room to students.

The login page comprises necessary contact address and other links displaying information related to the residences, the facilities and services available:

The Contact link: displays the contacts of the Residence Office and where users can put their comments.
Figure 1: Login Page

The *Residence* link: displays information about the residences (pictures, types of room, prices etc).

The *Facilities* link: shows the other facilities available at the residence (restaurant, cyber café, swimming pool, cinema etc).

The *Application* link: provides information about the online application procedure to help students to know one by one the process of application, requirements and time of application.
### 4.2.2 ACCOMMODATION APPLICATION PAGE

As shown on Fig 2. Firstly, the application page contains student details accessed from UWC database, these are: student number, name & surname, marital status, sex, address, cell phone number, email address etc. Secondly, the student can use a combo box to select a residence and option buttons to choose the type of room he/she wants to apply for, whether single or double. The Submit command button situated at the bottom of on the page allows the student to submit the application form after filling the required information.

![Accommodation Page](image-url)

<table>
<thead>
<tr>
<th>IDStudent</th>
<th>Name</th>
<th>SurName</th>
<th>Status</th>
<th>Address</th>
<th>Email</th>
<th>Telephone</th>
<th>Year</th>
<th>Sex</th>
<th>BirthDate</th>
</tr>
</thead>
<tbody>
<tr>
<td>2960564</td>
<td>2960564</td>
<td>mily</td>
<td></td>
<td>dddfdflf</td>
<td>gfgf</td>
<td>555</td>
<td>333</td>
<td>M</td>
<td>2009-04-06</td>
</tr>
</tbody>
</table>
4.2.3 Feedback Page

This is the page displayed as a feedback to the student after submission of the application. It provides a message to assure that the application was successful submitted and it contains a reference number for that application, the residence and room type applied for.

4.2.4 UWC Admin Page:

Figure 3: Feedback page

You are logged in as: toto.
Figure 4: UWC Administrator page

This page contains a text box enabling the Administrator to enter student’s reference number. The small command button helps to display the student detail (personal details, the residence, the room applied for and the charge for that room). Below there is a text box where the Administrator enters the amount paid and automatically the balance is calculated and displayed below.

4.2.5 ResMananger Page:

Figure 5: Residence administrator

The Residence Manager will be able to select the year of study and room type, then the system will automatically display all the students who applied for accommodation in that particular year and the list of rooms available. The residence manager can allocate room to students using the allocate link. S/He can search edit or delete information about room using room management link. The search names & manage occupancy link helps to retrieve information about rooms and students where s/he can replace or remove the students, and the reports link display the different reports such as the list of vacant rooms, list of occupied rooms, list of students and their rooms, etc.
4.3 How the User Interface behaves

The interface allows the users to input login username and password which must be authenticated before the access is granted. In case of wrong input of username or password, the application produces an error message prompting the user to input correct username or / and password. Also if the user input the correct username and password, the program informs the user that he has successfully logged in. the following user case diagram displays the functionality of the system.

![User Case Diagram]

Figure 6: Use case diagram
As shown in the above figure the student can view personal details, chooses the residence and room, and finally sends the application form. He/she can view and print reference number, retrieve, edit or cancel his application form. The administrator processes payments for each student who applies while the Residence Manager allocates room to student and both can retrieve, edit or cancel information.

4.4 How the user interacts with the system

4.4.1 HOW THE STUDENT INTERACTS WITH THE SYSTEM

Interactions between the student and the system are summarized in the following student activity diagram.
The online application is done after the student has read and agrees to the accommodation application information provided on the website. Also the list of activities below give an overview of how the student and interacts with the system in the process of the application.

1. The student will login into the Online Accommodation System by entering his/her student number and password on the login screen.

2. The system authenticate the student
3. The system will show the student’s details and application form

4. The student checks his details

5. The student choose room and send the application form

6. The system checks the room availability and correspondence between the room chosen and the student’s year of study.

7. The system provides the Student Application Reference Number.

8. Student prints the reference number

9. Student edit / cancel the application

10. The system update information

11. The system shows the updated information

12. The student logout

4.4.2 **HOW THE ADMINISTRATOR INTERACTS WITH THE SYSTEM**

Interaction between the administrator and the system is summarized in the following Administrator Activity Diagram.
Figure 8: Admin activity diagram

1. The administrator login
2. The system authenticates the user
3. The system displays the payment page
4. The administrator enters the reference number of the student
5. The system retrieves the student information from the database
6. The system displays the student information
7. The administrator enters the amount for payment

8. The administrator delete / edit / search information about payment

9. The system updates the information

10. The system shows the updated information

11. Logout.

4.4.3 **How the ResManager Interacts with the System**

Interaction between the ResManager and the system is summarized in the following ResManager Activity Diagram (see Figure X).

1. The ResManager login

2. The system authenticate the user

3. The system displays the main page

4. The ResManager selects the student’s year of study

5. The system displays the list of applicants in that year

6. The system shows the list of unoccupied rooms for that year

7. The ResManager allocated rooms to students

8. The ResManager search / edit / delete information

9. The system updates information

10. The system displays the information
11. The ResManager sends the confirmation to the student.

12. Logout

![ResAdmin activity diagram](image)

Figure 9: ResAdmin activity diagram

### 4.5 Conclusion

This chapter described the user interface. The log in page, main page and other pages’ functionalities were illustrated. The activity diagrams showed how the different types of users (Student, Administrator and ResManager) interact with the system. Furthermore, the
use cases showed what the users will do with the system. The next chapter will discuss the objects needed to implement the system.
Chapter 5

OBJECT ORIENTED ANALYSIS (OOA) OR HIGH LEVEL DESIGN

5.1 Introduction

The user interface was described in the previous chapter showing what the interface can do and how the user will interact with the system. In this chapter, each object is described and documented and a data dictionary provides the details of each object. Furthermore, the objects’ attributes and methods are identified.

5.2 Data Dictionary

The dictionary contains information describing the content of the Online Registration System.

5.2.1 Student Object

The Student object contains student personal information. This object contains one record for every student.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Data type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student Number</td>
<td>Student’s number given by the university</td>
<td>7 numbers</td>
<td>1234959</td>
</tr>
<tr>
<td>Title</td>
<td>Student’s title</td>
<td>20 Characters</td>
<td>Miss</td>
</tr>
<tr>
<td>Surname</td>
<td>Last name of the student</td>
<td>20 Characters</td>
<td>Sara</td>
</tr>
<tr>
<td>Name</td>
<td>First names of student</td>
<td>20 Characters</td>
<td>Mary Jane</td>
</tr>
<tr>
<td>Gender</td>
<td>Student’s gender coded M (Male), F (Female)</td>
<td>20 character</td>
<td>Female</td>
</tr>
<tr>
<td>DateBirth</td>
<td>Student’s birth date</td>
<td>Date</td>
<td>03-03-1990</td>
</tr>
<tr>
<td>Email</td>
<td>Student’s Email</td>
<td>20 Characters</td>
<td><a href="mailto:iraba@yahoo.fr">iraba@yahoo.fr</a></td>
</tr>
<tr>
<td>HomeAddress</td>
<td>Student’s address during holiday</td>
<td>20 Characters</td>
<td>19 Oakwood, Woodstock, CPT</td>
</tr>
<tr>
<td>Email</td>
<td>Student’s email address</td>
<td>20 Characters</td>
<td><a href="mailto:studuwc@hotmail.com">studuwc@hotmail.com</a></td>
</tr>
<tr>
<td>MobileNo</td>
<td>Student’s Cell phone number</td>
<td>Numbers</td>
<td>0724858774</td>
</tr>
<tr>
<td>Faculty</td>
<td>Student’s program of study</td>
<td>20 Characters</td>
<td>Engineering</td>
</tr>
<tr>
<td>Year Entry</td>
<td>The year the student entered the University</td>
<td>4 numbers</td>
<td>2002</td>
</tr>
</tbody>
</table>

Table 1:  Student Object

5.2.2  **Administrator Object**

The administrator object contains administrator personal information.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Data type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdminID</td>
<td>Administrator’s Identification number</td>
<td>7 numbers</td>
<td>0054959</td>
</tr>
<tr>
<td>DateBirth</td>
<td>Date of Birthday</td>
<td>Date</td>
<td>1970/12/12</td>
</tr>
<tr>
<td>Name</td>
<td>Administrator’s name</td>
<td>20 Characters</td>
<td>Mussa</td>
</tr>
<tr>
<td>Mobile Phone</td>
<td>Number of his/her phone</td>
<td>10 Number</td>
<td>078888800</td>
</tr>
</tbody>
</table>

Table 2:  Administrator Object
5.2.3 **ResAdmin Object**

The ResAdmin object contains ResAdmin’s personal information.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Data type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResAdmin ID</td>
<td>ResAdmin’s Identification number</td>
<td>7 numbers</td>
<td>0034959</td>
</tr>
<tr>
<td>DateBirth</td>
<td>Date of Birthday</td>
<td>Date</td>
<td>1945/1/14</td>
</tr>
<tr>
<td>Name</td>
<td>ResAdmin’s name</td>
<td>20Characters</td>
<td>Mussa</td>
</tr>
<tr>
<td>Mobile Phone</td>
<td>Number of his/her phone</td>
<td>10 Number</td>
<td>078844400</td>
</tr>
</tbody>
</table>

Table 3: ResAdmin Object

5.2.4 **Residence Object**

The Residence Object contains information about the residence of the UWC.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Data type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Residence Code</td>
<td>Code given to each residence</td>
<td>7 numbers</td>
<td>001</td>
</tr>
<tr>
<td>Residence Name</td>
<td>Actual Name of the residence</td>
<td>20Characters</td>
<td>Cassinga</td>
</tr>
<tr>
<td>NumberRoom</td>
<td>Number which are in residence</td>
<td>20Number</td>
<td>3000rooms</td>
</tr>
</tbody>
</table>

Table 4: Residence Object

5.2.5 **Room Object**

The Room Object contains information about a particular room of a given residence.
### Table 5: Room Object

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Data type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Room Code</td>
<td>Code given to each room</td>
<td>7 character</td>
<td>001/0001</td>
</tr>
<tr>
<td>ResidenceID</td>
<td>Actual code of the residence</td>
<td>characters</td>
<td>001</td>
</tr>
<tr>
<td>Room Type</td>
<td>The type of rooms in that residence</td>
<td>10 characters</td>
<td>Single</td>
</tr>
<tr>
<td>Price</td>
<td>The Charges per room</td>
<td>10 numbers</td>
<td>R 1864</td>
</tr>
<tr>
<td>Status</td>
<td>The status of room</td>
<td>10 characters</td>
<td>Vacant</td>
</tr>
</tbody>
</table>

### 5.2.6 PAYMENT OBJECT

The Payment Object contains information about a the payment for accommodation.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Data type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>PaymentID</td>
<td>Code of payment</td>
<td>7 character</td>
<td>001/0001</td>
</tr>
<tr>
<td>Student Number</td>
<td>Student Number of student who paid</td>
<td>10 Number</td>
<td>Vincent Biruta</td>
</tr>
<tr>
<td>Amount</td>
<td>Price of application fees</td>
<td>10 Number</td>
<td>R 500</td>
</tr>
<tr>
<td>Date</td>
<td>Date of payment</td>
<td>Date</td>
<td>02/03/2009</td>
</tr>
<tr>
<td>Amountpaid</td>
<td>Cash student paid</td>
<td>10 Number</td>
<td>R300</td>
</tr>
<tr>
<td>Balance</td>
<td>Rest payment</td>
<td>10 Number</td>
<td>R200</td>
</tr>
<tr>
<td>Reference Number</td>
<td>Number system display after application</td>
<td>5 Number</td>
<td>10005</td>
</tr>
</tbody>
</table>

Table 6: Payment object
### 5.2.7 BOOKING OBJECT

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Data type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>BookingID</td>
<td>Code given to each application made</td>
<td>7 character</td>
<td>45738</td>
</tr>
<tr>
<td>Student Number</td>
<td>Name of student who paid</td>
<td>10 characters</td>
<td>Vincent Biruta</td>
</tr>
<tr>
<td>Room Code</td>
<td>Room code applied for</td>
<td>10 Number</td>
<td>001/0001</td>
</tr>
<tr>
<td>Date</td>
<td>Date of application</td>
<td>Date</td>
<td>2009/11/11</td>
</tr>
</tbody>
</table>

**Table 7:** Booking Object

### 5.3 Class Diagram

The class diagram below shows the relationship between classes in the system. The main classes are student, administrator and ResAdmin. The diagram below shows that the student applies for a room and s/he pays for the room. The administrator controls and records the payments. The residence has rooms and rooms are managed by the ResAdmin. The ResAdmin also manages the applications and allocates rooms to successful applicants.
Figure 10: Class diagram
5.4 Conclusion

In this chapter, each class and its attributes were explained in detail using data dictionary. The data dictionary described the attributes and the data type. Also an example was given to each attribute. Furthermore, the class diagram demonstrated the relationship between classes in the system.
OBJECT ORIENTED DESIGN (OOD) OR LOWER LEVEL DESIGN

6.1 Introduction
In the previous chapter, each object was described and documented in data dictionary. However, in this chapter, the Object Oriented Design (OOD) will take the classes in the OOA a level deeper into the realm of pseudo-code. The OOD defines the data types for the attributes, the algorithms and implementation details of classes. Also the state and sequence diagrams of the system are explained.

6.2 Inner Details of Class Attributes and Methods
The class diagram described in chapter 5 shows all classes that will be used within the system. Also class attributes (data types) were defined in the data dictionary and methods (functions) are described within the class diagram.
6.3 State Diagram

The state diagram below depicts the dynamic behavior of login and application functions of the system. The student enters the username and password, and then the system validates the entries. If valid, the student is accepted into the system otherwise the student is rejected. For the student to be allocated a room, the system checks if the room is available, if not, the application is rejected. If the room is available the system will check whether the student has paid the room fee and has no outstanding balance (see Figure...)

![State Diagram](image_url)

Figure 11: State diagram
6.4  Sequence Diagram

The following diagram illustrates the sequence of activities as the system is into operation.

The student applies on application page, after that the system checks for room availability and makes confirmation to the application page. The application page displays the reference number to student. Then student make payment to the UWC Administrator and the Administrator input payment information and update the database. After that the ResAdmin selects room and students in the system, and the system displays the information about rooms and students’ applications. Thereafter the ResAdmin allocates rooms to students, update the database and send notifications to successful applicants.

![Sequence Diagram](image)

---

Figure 12:  Sequence diagram

---

36
6.5 Pseudo code

6.5.1 DISPLAY LOGIN PAGE

input "ID" and "Password" text boxes
if "LOGIN" button clicked then
{
    Search user in database for User's member if is not found then
    {
        displays error in page of login "User not found" start Procedure LOGIN again
    }
    Else
    {
        if username or password correct
        {
            if user = Student then
            {
                display Application Page
            }
            if user=Administrator then
            {
                System displays payment page
            }
        } else
        {
            display allocation Page
        }
    }
}

6.5.2 STUDENT APPLICATION

If student is not accepted in university
{
    display “You cannot login. You are not acceptable to continue your application.”
    then "Go Back" link clicked }
else
{
    display application Page which has all student details and where s/he can choose room
    if "submit" button clicked {
        if there is not the place{
            display error page with "no place of room in this residence try again"
        } else
        {
            Display the reference number
        }
    } }
Print the reference number and send to email

/*After event of “save” button the system display and send directly the reference number in email of student.*/
“save” button clicked
if there is room
    { select email from student and then put reference number
    }
After display the reference number
“print” button clicked{
Reference number is printed}

Edit and canceling the application

/*When the student want to modify or to canceling the application*/
"Search" button clicked{
search user database for student (by reference number ){
If it’s wrong reference number then {
display “incorrect reference number ”
then "Go Back" link clicked { go back to Student details page }
}
Else if "Edit" link clicked { Display Student details page } then
update information about room only
else {"Delete" link clicked { canceling the application }
}

6.5.3 PAYMENT

if user=Administrator then
{
System displays payment page
“input” reference number
Display the student details
“input” the amount
}

6.5.4 ALLOCATE

if user=ResAdmin then
{
//System displays allocate page
“select” student year of study from student Table

Echo 'the students details'
And
select" type_room from room Table
Echo 'the list of rooms which are unoccupied'}

if "SUBMIT" button clicked then { send email to student and update information in DB }
else
“RESET” button clicked then { modify the information }

6.3 Conclusion
This chapter tackled the pseudo-codes showed an outline of programs, written in a form which is understandable and which will be easy to convert into a programming language. It also covered the state diagram and sequence diagram.
REFERENCES

- Kevin Tatroe & Rasmus Lerdorf, Creating Dynamic Web Pages, Programming PHP, O’REILLY (http://books.google.co.za/books?id=7OjvOmOl3CcC&dq=php&printsec=frontcover&source=bl&ots=1oRkbZ44z)
- Definitions of terms of types of software testing, www.aptest.com/testtypes.html
BIBLIOGRAPHY

- Peter Ron & C. Coronac, Database system, Design, Implementation and Management
INDEX

A

After printing · 1

J

Junk · iv