

Governors State University OPUS Open Portal to University Scholarship

All Capstone Projects

Student Capstone Projects

Fall 2015


Online Dormitory Reservation System

Adithya Mothe
Governors State University

Koushik Kumar Suragoni
Governors State University

Ramya Vakity
Governors State University

Follow this and additional works at: <http://opus.govst.edu/capstones>

 Part of the [Databases and Information Systems Commons](#), and the [Systems Architecture Commons](#)

Recommended Citation

Mothe, Adithya; Suragoni, Koushik Kumar; and Vakity, Ramya, "Online Dormitory Reservation System" (2015). *All Capstone Projects*. 152.
<http://opus.govst.edu/capstones/152>

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to
http://www.govst.edu/Academics/Degree_Programs_and_Certifications/

Visit the [Governors State Computer Science Department](#)

This Project Summary is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact opus@govst.edu.

Table of Contents

1	<i>Project Description</i>	1
1.1	Project Abstract	1
1.2	Competitive Information	1
1.3	Relationship to Other Applications/Projects	1
1.4	Assumptions and Dependencies	3
1.5	Future Enhancements	3
1.6	Definitions and Acronyms	3
2	<i>Technical Description</i>	3
2.1	Project/Application Architecture	4
2.2	Project/Application Information flows	Error! Bookmark not defined.
2.3	Interactions with other Projects (if Any)	13
2.4	Interactions with other Applications	13
2.5	Capabilities	13
2.6	Risk Assessment and Management	13
3	<i>Project Requirements</i>	14
3.1	Identification of Requirements	14
3.2	Performance Requirement	14
3.3	Security and Fraud Prevention	15
3.4	Software and Hardware Requirements	15
3.5	HTML and JavaScript	16
4	<i>Project Design Description</i>	20
5	<i>Project Internal/external Interface Impacts and Specification</i>	23
6	<i>Project Design Units Impacts</i>	23
6.1	Functional Area/Design Unit A	23
6.1.1	<i>Functional Overview</i>	25
6.1.2	<i>Impacts</i>	26
6.2	Functional Area/Design Unit B	26
6.2.1	<i>Functional Overview</i>	26
6.2.2	<i>Feasibility Report</i>	27
6.2.3	<i>System Testing and Implementation</i>	29
6.3	Output Screenshots	32
7	<i>Open Issues</i>	32
8	<i>Acknowledgements</i>	32
9	<i>References</i>	32
10	<i>Appendices</i>	32

1 PROJECT DESCRIPTION

1.1 Project Abstract

This project is Online Dorms Systems which allows users to book their room in the dorm from anywhere; this is an automated system where the user can search the availability of rooms in the dorm.

The search can be done based on the dates. The rooms that available are come with the status available, it will display all the rooms available as of that particular search date. Once the room has been booked the user can cancel the reservation within 48 hours. And there is concept of user login. As the user creates his own account with his email id, he can login into his account.

There is an administrator login page for administrator, who has the control over the dorms system application. He can also review the business profit and growth. The proposed system reduces a lot effort for student and the administrator of the dorms as well.

The objective of this application is to develop an online dorm system for Improving Software Quality and Reliability is useful for applications developed in an organization. This system can be used for reduce problems of student against an application/module, assigning dorms to individuals and solving problems of student.

1.2. Competitive Information

The objective of this application is to develop an online dorm system for Improving Software Quality and Reliability is useful for applications developed in an organization. This system can be used for reduce problems of student against an application/module, assigning dorms to individuals and solving problems of student.

1.3 Relationship to other applications

The existing system in does not contain any online facility to book the dorms remotely. All they have in our University is the applicant, who is willing to book or reserve a bed in the dorms have to go and meet the dorms administrator in personnel and have to fill up application form so that the administrator can review the applicant's information and then administrator allocates a bed in dorm. This all process goes only on paper based. The applicant can see like how many beds or rooms are available in the dorms. If the applicant vacates ones and if want to come back and book the room again he, then he has to go through all the procedure explained earlier. When the applicant make a reservation i.e., pays the bill and he does not want to the reserved room because of his personal reasons, he will not get his money back at any cost. It is a kind of loss to the applicant. And our system has a main drawback that does not show any reserved and vacant beds when the beds are booked.

To avoid and overcome this problem, I want develop a system in which the user or the applicant can cancel his reservation within 48 hours of time, with an option of money back. If the exceeds 48 hours then he cannot get the money back.

In the existing system the applicant has to pay the fee for the bed has to pay at the administrator's office. Where as in the proposed system the applicant can select and reserve the room and pay the bill online. Here, in this existing system we have many drawbacks and tired of filing the application manually, approving and finally to allot

the bed Main problem of the existing system is the only paper based work and everything has to do it physically and they should pass the documents by hand to hand from starting and there is no concept of the information reusability.

Drawbacks of the Existing System

As I discussed earlier in this chapter about the existing system which is using in the university takes a lot of time to start each on each applicant request to allot the rooms. Mainly it consuming a lot of time because of only thing is paper based work and whenever the applicant visits second it would be like waste of time for both the applicant and the administrator as well.

Mainly in this paper based work we don't have any kind of storage system to access the information of the applicant whenever the administrator wants. Here, if any applicant's documents missed in the administrator's office unfortunately then the applicant have come again and do all the application process again. In this case the dorms management may lose the faith and credibility among their customers. So, this makes me to generate a system which can store a lot of data and can be able to access in future time, by doing this I can solve the problems like lack of information reusability and work effort in case of the lost application.

When security comes into picture it makes a lot of difference to the current system because it's only the paper based work so we can't provide any kind of security to the existing system. Some of the applicants may do not want their information like phone number and address go publicly this. At this time, may some other people in the admission room can see the information about the applicant. But, when I got the thought of creating an online automated system myself took a lot security problem in order to provide to the proposed system.

As there is no online system so the applicant has to pay the bill at the administrator's office only. In the proposed system there should be a mechanism in which the applicant can pay the bill online.

Need of the System:

The system itself tells that it is the online method of traditional dorm reservation system, by developing this system there would not any functional changes but there will be increase in performance and functionality. Sending the applicants information through online gives the solution to some of the problems that are encountered with the existing system. In this project, the security aspects are evaluated.

By using the proposed system we can achieve the following benefits:

- User can enjoy the cool and comfort User Interface.
- The applicant does not have to come to the administrator's office to make a reservation, so that the effort for the applicant can be reduced.
- There will be an account for each and every user which validates them based on their Email address. So that the applicant does not have to give their information as many times as they visit.
- The information reusability has achieved.
- The user data can be stored in a database.
- The administrator will have an account through which he can login and know the business statistics like revenue and profit.

The challenges that have faced initially when I thought of the proposed system are:

- Have to provide a feel good and user friendly User Interface so that the applicant feel comfort than that of the existing system.

Have to provide a system, in which the applicant's information can be used in the future, i.e. the applicant has to have a profile or account with his information in it.

- Have to provide a secure pay method as my system deal with the money payment.
- Have to design a data storage system that stores as many applicant's details as possible.
- Have to provide a safe and secure money payment method. In case he cancels the reservation the system has to provide the applicant with a money back option.

1.4 Assumptions and Dependencies

The proposed system organizes the data effectively. The online dorms system for improving application Quality and Reliability is a web based application that can be accessed throughout the organization. It can be useful application that can be taken as reference for any kind of situation or project.

1.5 Future Enhancements

We can provide a full- fledged user account and we can also give notifications for that account. We can also add events to the system that will let us know if anything going to happen in the dorms. Also based on this web application we can also create some smartphone applications.

1.6 Definitions and Acronyms

SRS – Software Requirement Specifications

MMS – Meeting Scheduler System

FR – Functional Requirement

NFR –Nonfunctional requirement

JDBC – Java Database Connectivity

HTTP- Hypertext transfer Protocol

HTML- Hypertext Markup Language

2 Technical Description

As discussed earlier, there are certain problems with the current dorm management system in the university. Sometimes the application may be lost completely without even knowing to the administrator and the people who are participating in the application due to the negligence of the either the administrator or the people working under the administrator. And there is no security for the user or applicant's information. These are the factors to motivate me to develop this project. Determination of whether the Dorms management is worth doing or not is done in this section. Finally I found the result positive.

2.1 Project / Application Architecture:

The project has been developed in Model-2 architecture. Model 2 is a complex design pattern used in the design of Java Web applications which separates the display of content from the logic used to obtain and manipulate the content. Since Model 2 drives a separation between logic and display, it is usually associated with the model–view–controller (MVC) paradigm. While the exact form of the MVC "Model" was never specified by the Model 2 design, a number of publications recommend a formalized layer to contain MVC Model code. The Java Blue Prints, for example, originally recommended using EJBs to encapsulate the MVC Model.

In a Model 2 application, requests from the client browser are passed to the controller. The controller performs any logic necessary to obtain the correct content for display. It then places the content in the request (commonly in the form of a JavaBean or POJO) and decides which view it will pass the request to. The view then renders the content passed by the controller. Model 2 is recommended for medium- and large-sized applications.

The following figure shows a typical Model2 architecture. In which a database and a web browser can communicate using MVC (Model-View-Control)

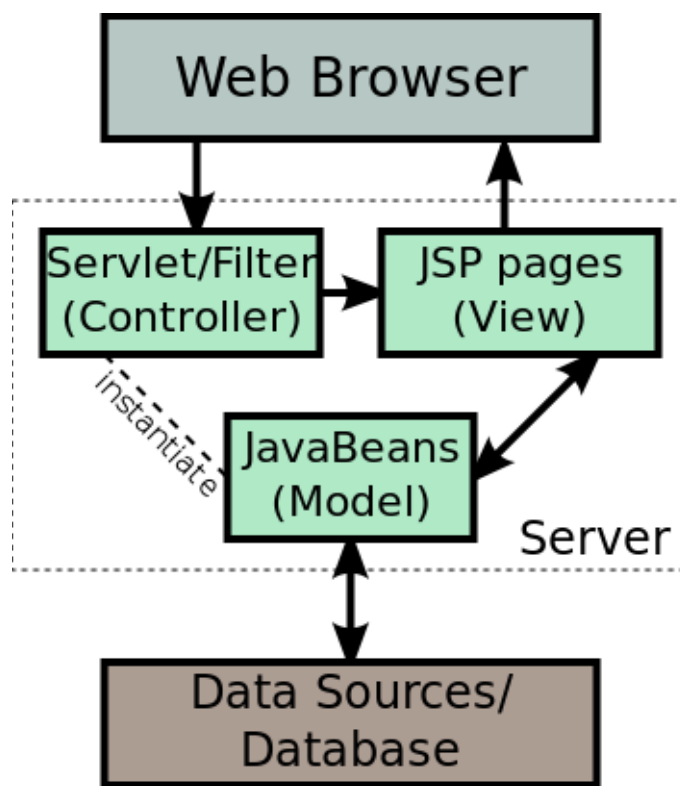


Fig1: A typical Model 2 architecture

The History of the Model 2 architecture:

In 1998, Sun Microsystems published a pre-release of the Java Server Pages specification, version 0.92. In this specification, Sun laid out two methods by which JSP pages could be used. The first model (referred to as

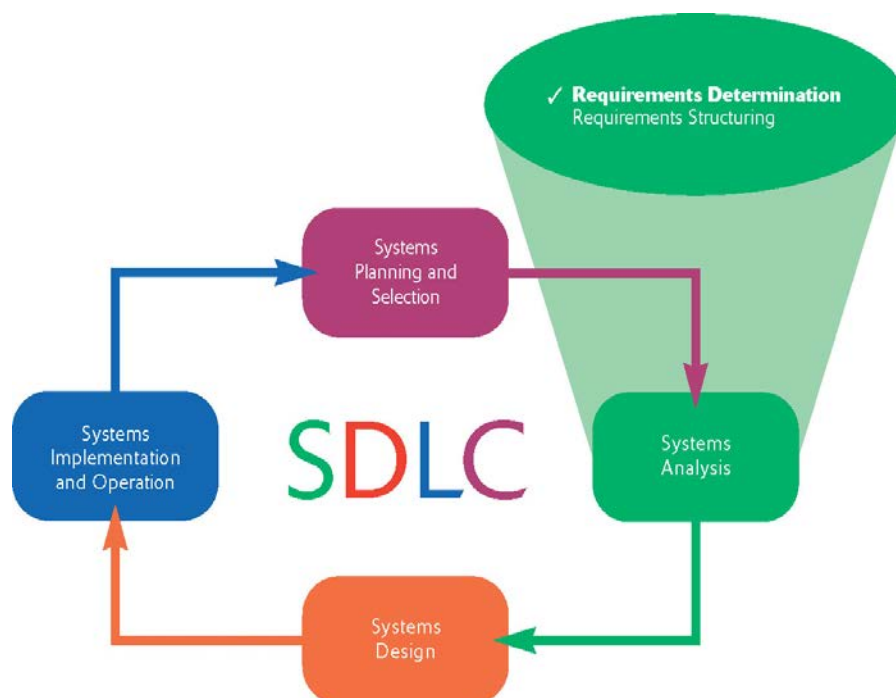
"model 1" due to its ordering in the document) was a simplistic model whereby JSP pages were standalone, disjointed entities. Logic could be contained within the page itself, and navigation between pages was typically achieved by way of hyperlinks. This fit with the then-common usage of template technology. ColdFusion and Active Server Pages are examples of contemporary technologies that also implemented this model.

The second model referred to by the document ("model 2" in the ordering) was an improved method that combined servlet technology with JSP technology. The specific difference listed was that a servlet would intercept the request, place the content to render into a request attribute (typically represented by a JavaBean), then call a JSP to render the content in the desired output format. This model differed from the previous model in the fact that JSP technology was used as a pure template engine. All of the logic was separated out into a servlet, leaving the JSP with the sole responsibility of rendering the output for the content provided.

In March 2000, the Apache Struts project was released. This project formalized the division between View and Controller and claimed implementation of the "Model 2" pattern. Once again, the implementation of the "Model" was left undefined with the expectation that software developers would fill in an appropriate solution. Database interaction via JDBC and EJBs were options suggested on the Struts homepage. More recently, Hibernate, iBati and Object Relational Bridge were listed as more modern options that could be used for a model.

Since the release of Struts, a number of competing frameworks have appeared. Many of these frameworks also claim to implement "Model 2" and "MVC". In result, the two terms have become synonymous in the minds of developers. This has led to the use of the term "MVC Model 2" or "MVC2" for short.

Process Model Used With Justification



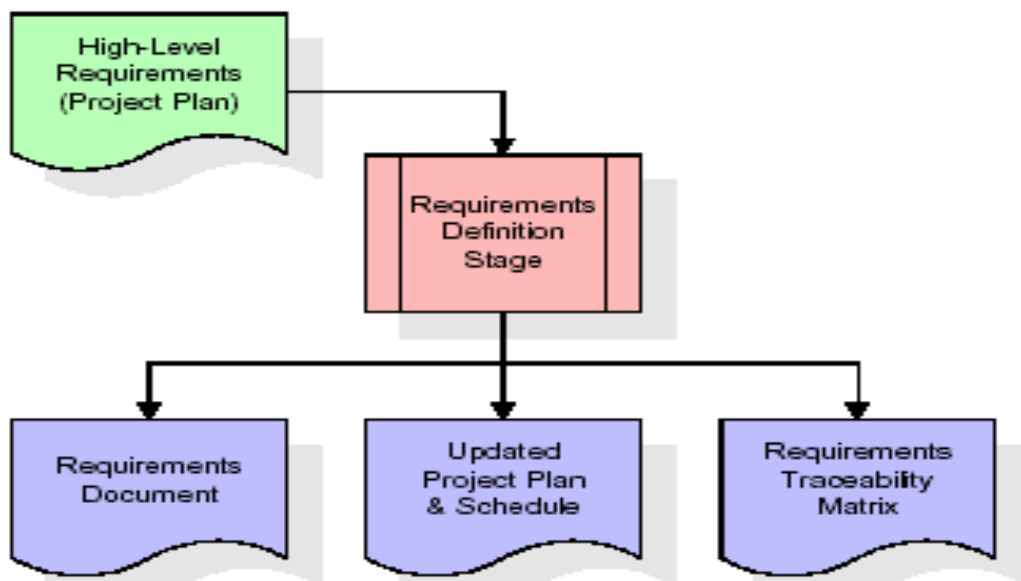
Stages in SDLC:

- Requirement Gathering
- Analysis
- Designing
- Coding
- Testing
- Maintenance

Requirements Gathering stage:

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define

Operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

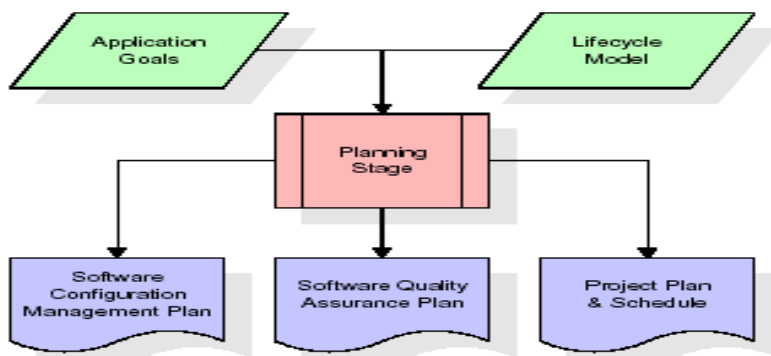
In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

- Feasibility study is all about identification of problems in a project.
- No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
- Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator

Analysis Stage:

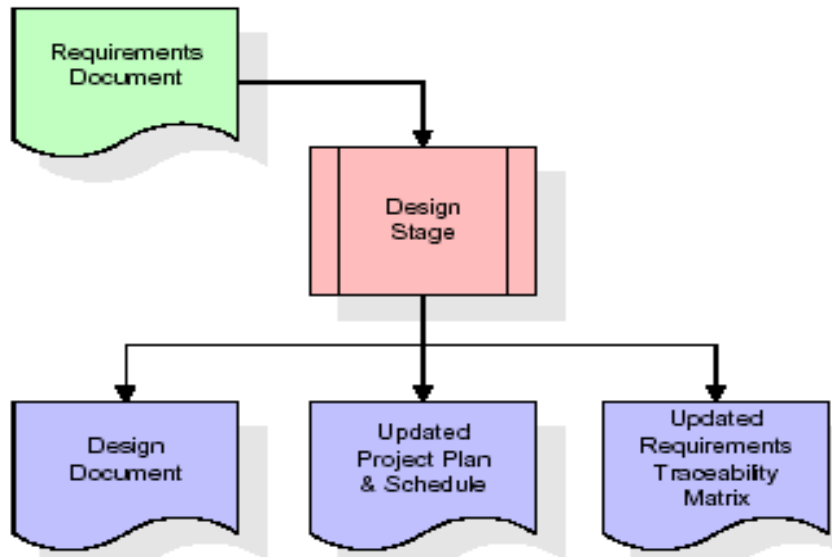
The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

Designing Stage:

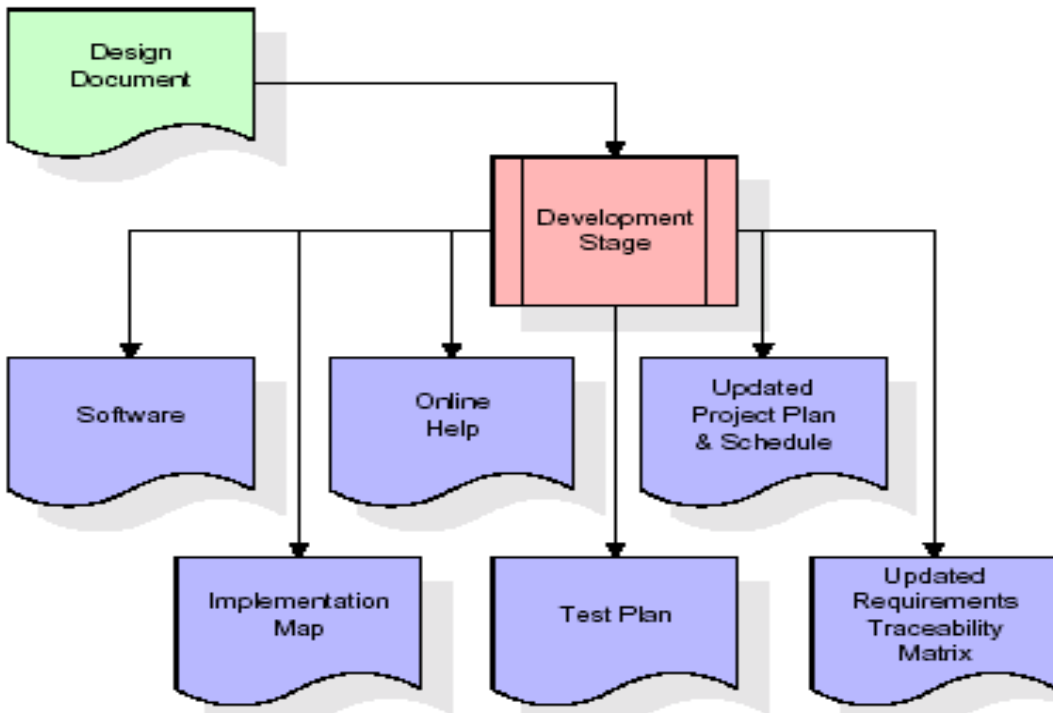
The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.



When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

Development (Coding) Stage:

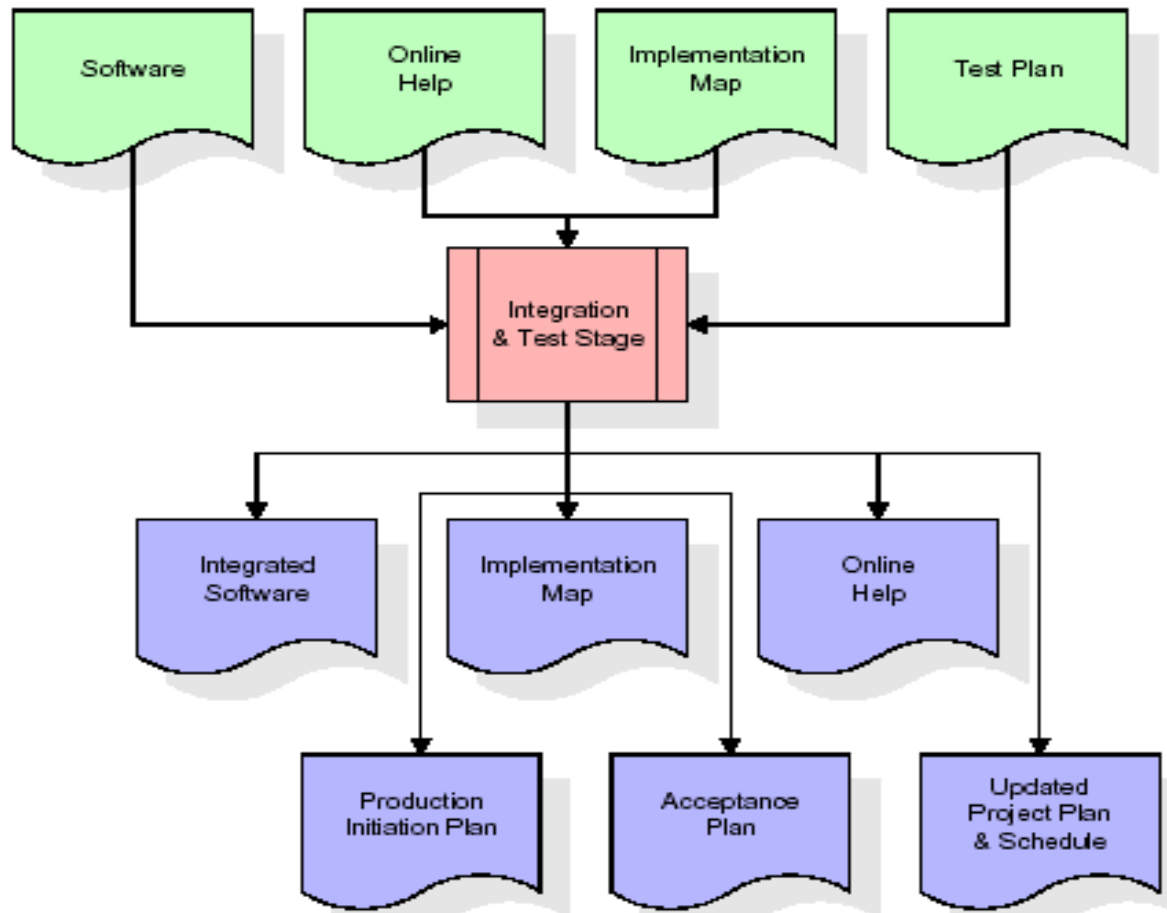
The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

Integration & Test Stage:

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

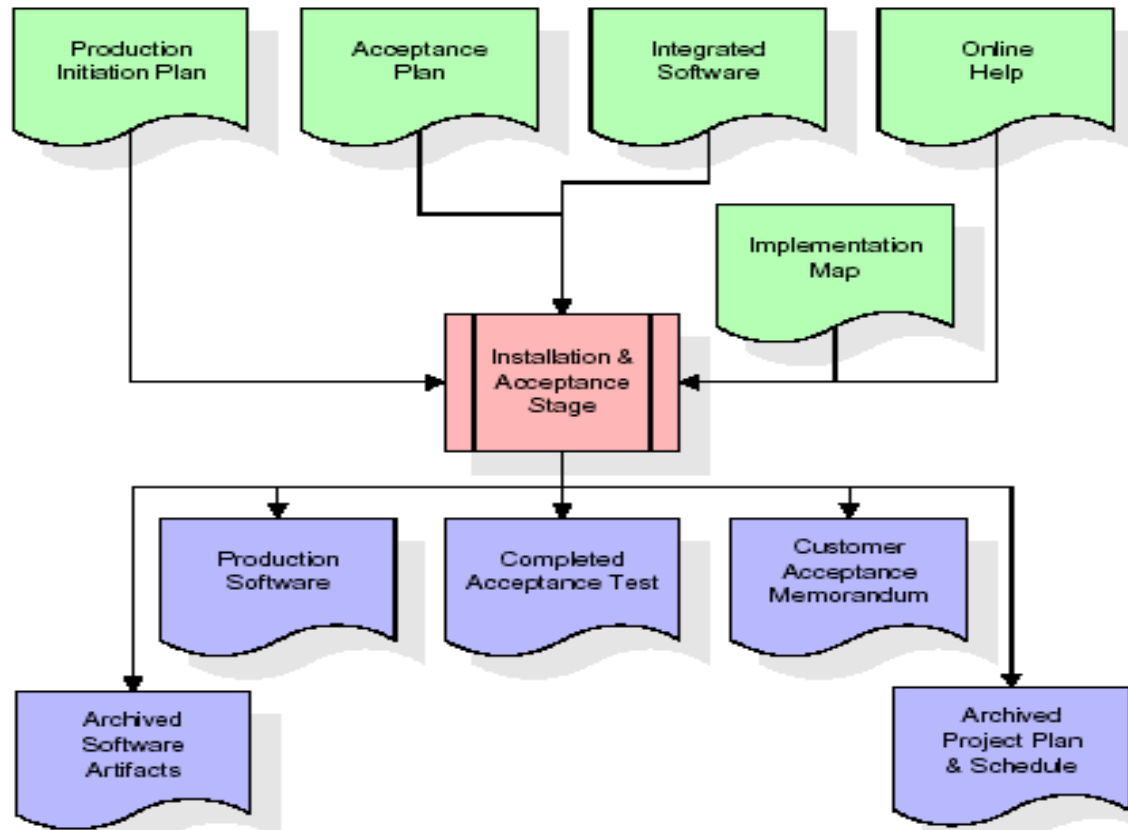


The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

Installation & Acceptance Test:

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

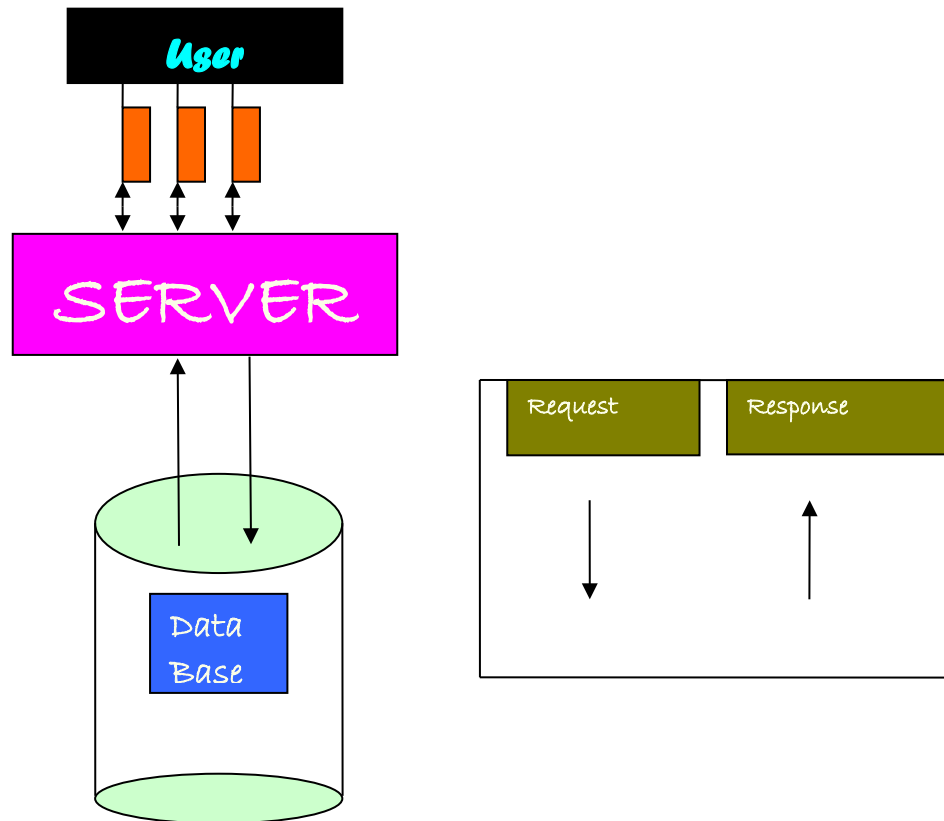
Maintenance:

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category.

For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks)

2.2. Project/ Application Information flow:

Below architecture diagram represents mainly flow of requests from users to database through servers. In this scenario overall system is designed in three tiers separately using three layers called presentation layer, business logic layer and data link layer. This project was developed using 3-tier architecture.



2.3 Interaction with other Projects

In addition to dividing the application into three kinds of components, the Model–view–controller (MVC) design defines the interactions between them.

- A controller can send commands to the model to update the model's state (e.g., editing a document). It can also send commands to its associated view to change the view's presentation of the model (e.g., by scrolling through a document).

A model notifies its associated views and controllers when there has been a change in its state. This notification allows the views to produce updated output, and the controllers to change the available set of commands. A passive implementation of MVC omits these notifications, because the application does not require them or the software platform does not support them.

2.4 Interactions with other Applications

We have took this project from a reservation system and thought of assigning it to the dorms. Took references form our GSU Hostel reservation system. Our project adds a new feature when it come to our universities reservation application system. It will give an option to reserve the room and we can hold it for 48 hours. If the room did not booked in 48 hours then the admin has the ability to cancel the reservation.

2.5 Capabilities

The architecture of this project includes several individual modules. While developing the project each module has developed individually. The following figure shows the process flow of all the modules involved in the project.

2.6 Risk Assessment and Management

Before Risk assessment starts it is basic that an establishment is built up for giving organized venture data, therefore, the accompanying task components were finished and characterized preceding building up this Risk Management Plan:

- Define the schedule, resources and cost for the project.
- Develop a blue print for the Project
- Develop master schedule and detailed schedules
- Estimate the budget for the project
- Identify required and available resources
- Define Risk Management Roles and Responsibilities
- Establish performance measurement metrics
- Risk management should be handled.

Project group takes part in risk assessment gatherings and individuals serve as meeting recorder and timekeeper Key partners partake in danger evaluation gatherings Project Sponsor may take part in danger appraisal gatherings.

3. REQUIREMENT SPECIFICATIONS

3.1 Identification of Requirements

The first phase of the project is the requirement specification. We have divided this phase into 3 modules.

The 3 modules in the project are,

- UI Creation and Database Design
- User Module
- Admin Module

UI Creation and database Design:

In the initial phase of the implementation the User Interface has to be designed, which makes user feel comfortable. It will support the user to check the availability of the beds according to the dates. Once UI design completed, then main focus is on database design. The database will be designed with all the required tables and the relationships derived.

User Module:

The second module in the project is to satisfy all the user's requirements. This module starts with designing a user account for the respective user. In this module user can register with the System by giving his personal information. Once the registration is completed the user can login in to the system and can check for the availability of the beds.

Admin Module:

In this module the administrator's activities are developed. The administrator can add a bed or delete a bed. He has all the control over the system. The administrator can login with his email address and password. Here the password is encrypted and stored in the database using some encryption algorithms. While retrieving the password to login the system decrypts the password using respective decryption algorithms.

3.2 Performance Requirements

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

3.3 Security and Fraud Prevention:

Security is the main thing that we need to take care of. In our project, we are using a secure gateway for the payment. It will automatically connects to the concerned bank and the payment process will continue. For the security we are using this secured gateway for the payment. The same thing will be applied to the fraud prevention. We are using the security measures for the frauds to not to interrupt the payment process.

System Security:

Setting Up Authentication for Web Applications

Introduction:

To configure authentication for a Web Application, use the `<login-config>` element of the `web.xml` deployment descriptor. In this element you define the security realm containing the user credentials, the method of authentication, and the location of resources for authentication.

Security in Software

To set up authentication for Web Applications:

1. Open the `web.xml` deployment descriptor in a text editor or use the Administration Console. Specify the authentication method using the `<auth-method>` element. The available options are:

BASIC

Basic authentication uses the Web Browser to display a username/password dialog box. This username and password is authenticated against the realm.

FORM

Form-based authentication requires that you return an HTML form containing the username and password. The fields returned from the form elements must be: `j_username` and `j_password`, and the action attribute must be `j_security_check`. Here is an example of the HTML coding for using FORM authentication:

```
<form method="POST" action="j_security_check">
  <input type="text" name="j_username">
  <input type="password" name="j_password">
</form>
```

The resource used to generate the HTML form may be an HTML page, a JSP, or a servlet. You define this resource with the `<form-login-page>` element.

3.4 Software and Hardware Requirements:

Software Requirements:

Technology	: Java and J2EE
Web Technologies	: HTML, CSS, JavaScript, JDBC, JSP

Database	: mysql
IDE	: Eclipse juno
JDK Version	: JDK1.7
Server	: Tomcat5.5

Hardware Requirements:

Processor	: Pentium
RAM	: 1GB

3.5 HTML and JavaScript:

Hyper Text Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces web pages that included text, graphics and pointer to other web pages.

HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language),but Specialized to hypertext and adapted to the Web. We can navigate through the information based on out interest and preference. A markup language is simply a series of items enclosed within the elements should be displayed.

Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.Html can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop HTML provides tags (special codes) to make the document look attractive.

HTML provides are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

Basic Html Tags:

<!-- -->	Specific Comments.
<A>.....	Creates Hypertext links.
.....	Creates hypertext links.
<Big>.....</Big>	Formats text in large-font
<Body>.....</Body>	Contains all tags and text in the Html-document
<Center>.....</Center>	Creates Text
<DD>.....</DD>	Definition of a term.
<TABLE>....</TABLE>	Creates table
<Td>.....</Td>	indicates table data in a table.
<Tr>.....</Tr>	designates a table row
<Th>.....</Th>	creates a heading in a table.

• Advantages:

- A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
- HTML is platform independent

- HTML tags are not case-sensitive
- **Java Script**

JavaScript is a compact, object-based scripting language for developing client and server internet applications. Netscape Navigator 2.0 interprets JavaScript statements embedded directly in an HTML page. And Livewire enables you to create server-based applications similar to common gateway interface (cgi) programs.

In a client application for Navigator, JavaScript statements embedded in an HTML Page can recognize and respond to user events such as mouse clicks form Input, and page navigation.

For example, you can write a JavaScript function to verify that users enter valid information into a form requesting a telephone number or zip code. Without any network transmission, an Html page with embedded Java Script can interpret the entered text and alert the user with a message dialog if the input is invalid or you can use JavaScript to perform an action (such as play an audio file, execute an applet, or communicate with a plug-in) in response to the user opening or exiting a page.

4 PROJECT DESIGN DESCRIPTION

We have used Java and JSP pages for the front end and MYSQL for the backend. Java uses integration methods to create the application and MYSQL has the secure system that will store the data automatically as we have used the Model 2 architecture.

Online Dormitory Reservation System, provides some advance methods for the users. We have taken some specific input and output measures in the design of the document to provide a user friendly and reliable reservation system.

To provide flexibility to the users, the interfaces have been developed that are accessible through a browser. The GUI'S at the top level have been categorized as

1. Administrative user interface
2. The operational or generic user interface

The 'administrative user interface' concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. These interfaces help the administrators with all the transactional states like Data insertion, Data deletion and Date updating along with the extensive data search capabilities.

The operational or generic user interface' helps the end users of the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information in a customized manner as per the included flexibilities

- **Input & Output Representation**

Input design is a part of overall system design. The main objective during the input design is as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

Input Stages:

The main input stages can be listed as below:

- recording
- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission
- Data validation
- Data correction

Input Types:

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.
- Operational, which are computer department's communications to the system?
- Interactive, which are inputs entered during a dialogue.

Input Media:

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to:

- Flexibility of format
- Type of input
- Accuracy
- Speed
- Verification methods
- Rejection rates
- Ease of correction
- Storage and handling requirements
- Security
- Easy to use

- Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

Output Design:

In general are:

- External Outputs whose destination is outside the organization.
- Internal Outputs whose destination is within organization and they are the User's main interface with the computer. Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly with the system.

Output Definition

1 The outputs should be defined in terms of the following points:

- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

For Example

- Will decimal points need to be inserted
- Should leading zeros be suppressed.

Output Media:

In the next stage it is to be decided that which medium is the most appropriate for the output. The main considerations when deciding about the output media are:

- The suitability for the device to the particular application.
- The need for a hard copy
- The response time required.

- The location of the users
- The software and hardware available.

Keeping in view the above description the project is to have outputs mainly coming under the category of internal outputs. The main outputs desired according to the requirement specification are:

The outputs were needed to be generated as a hard copy and as well as queries to be viewed on the screen. Keeping in view these outputs, the format for the output is taken from the outputs, which are currently being obtained after manual processing. The standard printer is to be used as output media for hard copies.

5. PROJECT INTERNAL/ EXTERNAL INTERFACE IMPACTS AND SPECIFICATIONS

Database Design:

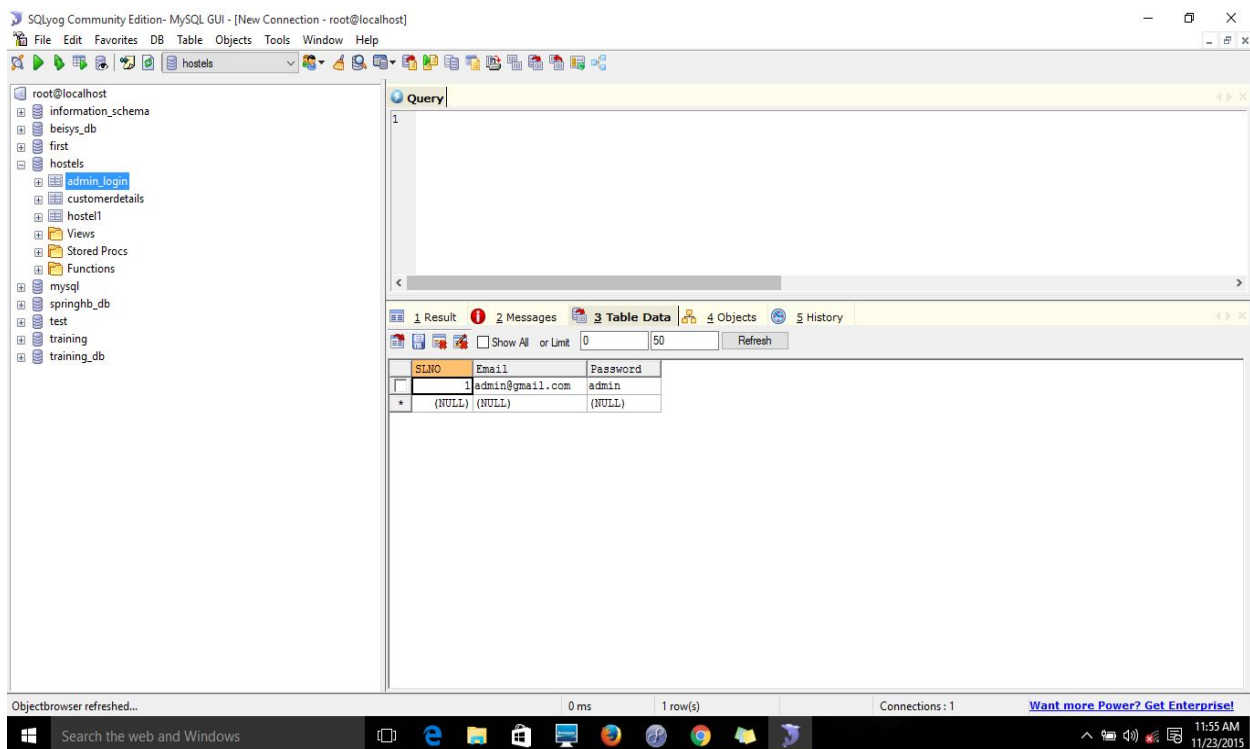
We have used MySQL database to store the data. This database can be run on internet and can also be on our local host, we have used xampp control panel which holds both MySQL database and tomcat server.

Initially we have created database with names “records” and “withdrawal dB” for equipment request and withdrawal form respectively.

Each data base includes several tables, records database includes two tables named report1, which holds the information about the professor who has requested for equipment and all about that request. It also contains user login table holds the username and password of the administrator, when administrator tries to log on the portal it checks with the database for authentication.

The following figures shows the Tables in the database.

Table 1 - Admin Login



The screenshot displays the SQLyog Community Edition interface. On the left, the 'hostels' database is expanded, showing several tables including 'admin_login'. The 'admin_login' table is selected, and its data is displayed in the 'Table Data' pane on the right. The data shows a single row with the following values:

SLNO	Email	Password
1	admin@gmail.com	admin
*	(NULL)	(NULL)

The status bar at the bottom indicates '0 ms' and '1 row(s)'.

Table 2 – Customer Details

SQLyog Community Edition- MySQL GUI - [New Connection - root@localhost]

File Edit Favorites DB Table Objects Tools Window Help

root@localhost

- information_schema
- beisys_db
- first
- hostels
 - admin_login
 - customerdetails
 - hostel1
 - Views
 - Stored Procs
 - Functions
- mysql
- springhb_db
- test
- training
- training_db

Query

1

1 Result 2 Messages 3 Table Data 4 Objects 5 History

Show All or Limit 0 50 Refresh

SLNO	FirstName	MiddleName	LastName	Email	Password	PhoneNumber	Address1
1	null	null	null	null	'2&BSX 67&+&#	(null)null-null	null
2	AAA		MM	aaa@gmail.com	'ia&kIIZ&i+*3ne&	(312)208-3145	chicago
3	Adithya		Mothe	mothe.adithya@gmail.com	-	(312)208-3145	2951 s king
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Objectbrowser refreshed... 0 ms 3 row(s) Connections: 1 Want more Power? Get Enterprise!

Search the web and Windows 11:56 AM 11/23/2015

Table 3 - Hostel

SQLyog Community Edition- MySQL GUI - [New Connection - root@localhost]

File Edit Favorites DB Table Objects Tools Window Help

root@localhost

- information_schema
- beisys_db
- first
- hostels
 - admin_login
 - customerdetails
 - hostel1
 - Views
 - Stored Procs
 - Functions
- mysql
- springhb_db
- test
- training
- training_db

Query

1

1 Result 2 Messages 3 Table Data 4 Objects 5 History

Show All or Limit 0 50 Refresh

SLNO	Room	Bed	DatesAvailable	Cost	Status	Customer_Email	Date/TimeOfSubmission
1	1	1	11-01-2015	12	Vacant		2015-11-23 10:46:56
2	1	2	11-01-2015	12	Vacant		2015-11-23 10:55:37
3	2	3	11-25-2015	12	Vacant		2015-11-23 10:54:30
4	2	4	11-01-2015	12	Vacant		2015-11-23 10:54:33
5	3	5	11-01-2015	12	Vacant		2015-11-23 10:54:35
6	3	6	11-01-2015	12	Vacant		2015-11-23 10:55:17
7	1	7	11-01-2015	13	Vacant		2015-11-23 10:54:39
8	1	8	11-01-2015	13	Vacant		2015-11-23 10:54:41
9	4	9	11-01-2015	20	Vacant		2015-11-23 10:54:43
10	5	10	11-01-2015	20	Vacant		2015-11-23 10:55:05
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	CURRENT_TIMESTAMP

Objectbrowser refreshed... 0 ms 10 row(s) Connections: 1 Want more Power? Get Enterprise!

Search the web and Windows 11:57 AM 11/23/2015

Normalization

A Database is a collection of interrelated data stored with a minimum of redundancy to serve many applications. The database design is used to group data into a number of tables and minimizes the artificiality embedded in using separate files. The tables are organized to:

- Reduced duplication of data.
- Simplify functions like adding, deleting, modifying data etc.,
- Retrieving data
- Clarity and ease of use
- More information at low cost

Normalization

Normalization is built around the concept of normal forms. A relation is said to be in a particular normal form if it satisfies a certain specified set of constraints on the kind of functional dependencies that could be associated with the relation. The normal forms are used to ensure that various types of anomalies and inconsistencies are not introduced into the database.

First Normal Form:

A relation R is in first normal form if and only if all underlying domains contained atomic values only.

Second Normal Form:

A relation R is said to be in second normal form if and only if it is in first normal form and every non-key attribute is fully dependent on the primary key.

Third Normal Form:

A relation R is said to be in third normal form if and only if it is in second normal form and every non key attribute is non-transitively depend on the primary key

Impacts

By following the specifications we can get an idea about what we need to do and by following the database design and normalizations we will find out the internal and external interface impacts and how it effects the project. Mainly we need to focus on the database design and the front end can be done by following the design requirements we will get an output which will suit our project.

6. PROJECT DESIGN AND UNITS IMPACTS

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap and synergy with the disciplines of systems analysis, systems architecture and systems engineering.

6.1 Functional Area/Design Unit A

6.1.1Functional Overview:

Mainly the project design follows the Waterfall model, we need to find the requirements first and then we should go to the other phases like construction and deployment. It can be described by UML Diagrams and Activity Diagrams.

UML DIAGRAMS

Unified Modeling Language:

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

- User Model View
 - i. This view represents the system from the user's perspective.
 - ii. The analysis representation describes a usage scenario from the end-users perspective.
- Structural model view
 - i. In this model the data and functionality are arrived from inside the system.
 - ii. This model view models the static structures.
- Behavioral Model View

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.
- Implementation Model View

In this the structural and behavioral as parts of the system are represented as they are to be built.
- Environmental Model View

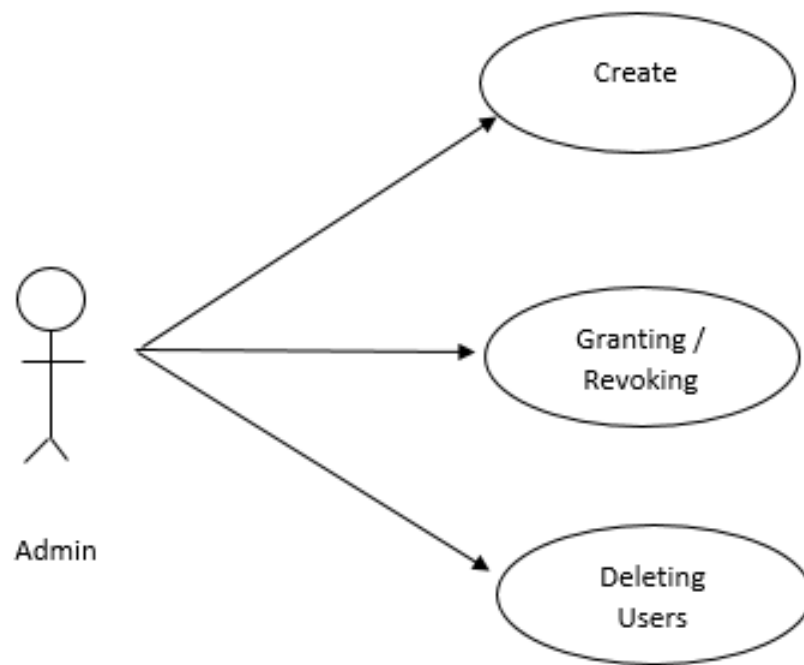
In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the system.
- UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

Use case Diagrams represent the functionality of the system from a user's point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.

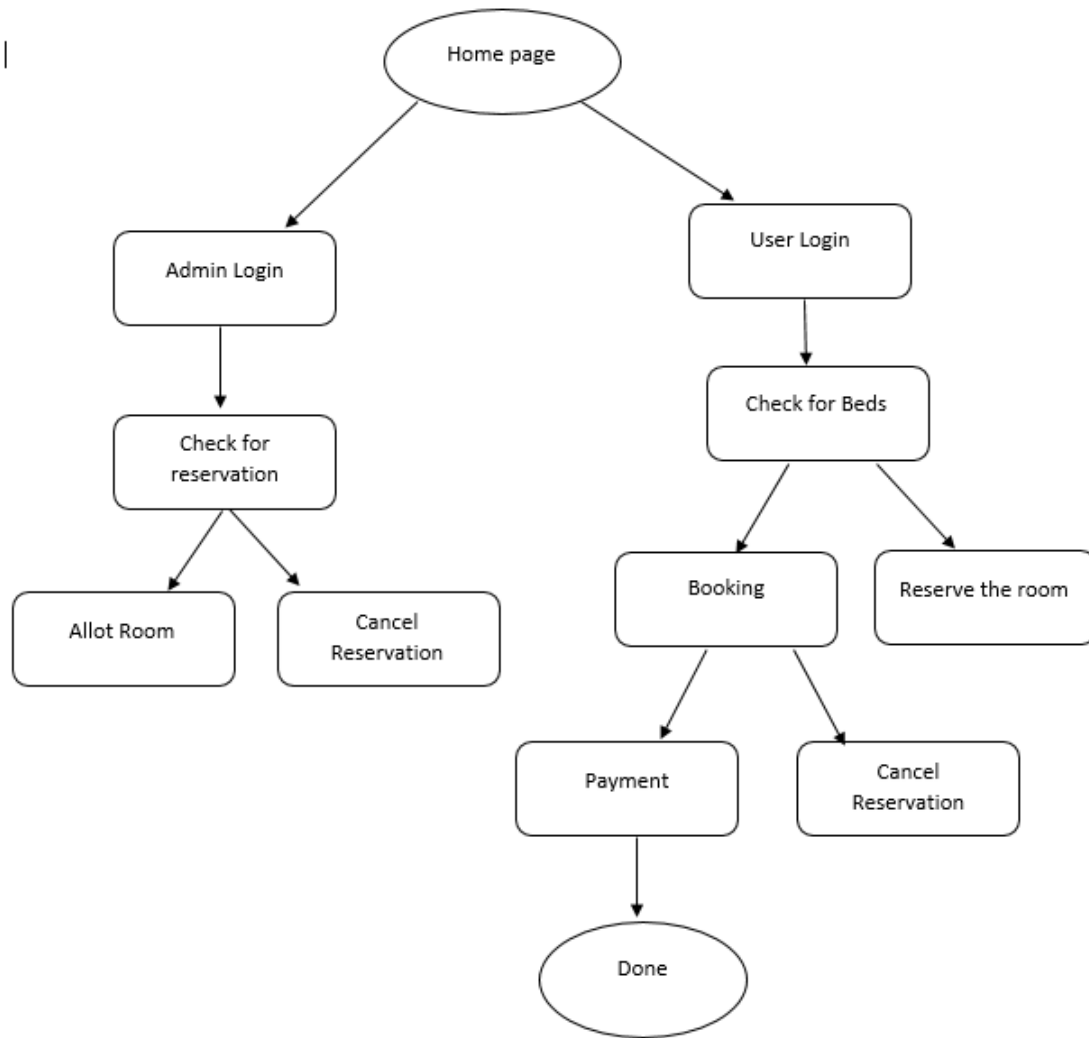
Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer ...etc., or another system like central database.



User Module: Use case diagram

Activity Diagram:

Activity Diagram describes the complete flow of the process. It will clearly describes how the process and control flow will takes place. As per the diagram everything starts from the Home page. The User and Admin can login and do play the respective roles like booking, reserving and cancelling the reserving beds.



Activity Diagram

6.1.2. Impacts:

By following the above processes we can get to know the complete flow of the processes. By watching the activity diagram we can get to know everything in the project. If there is any issue then we can easily find out by following the UML and Activity diagrams.

6.2 Functional Area/Design Unit B

6.2.1 Functional Overview

6.2.2 Feasibility Report

Preliminary investigation examines project feasibility; the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operation Feasibility
- Economic Feasibility

Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment's have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?

Are there technical guarantees of accuracy, reliability, ease of access and data security?

Operational Feasibility

User-friendly

Customer will use the forms for their various transactions i.e. for adding new routes, viewing the routes details. Also the Customer wants the reports to view the various transactions based on the constraints. These forms and reports are generated as user-friendly to the Client.

Reliability

The package will pick-up current transactions on line. Regarding the old transactions, User will enter them in to the system.

Security

The web server and database server should be protected from hacking, virus etc.

Portability

The application will be developed using standard open source software (Except Oracle) like Java, tomcat web server, Internet Explorer Browser etc. these software will work both on Windows and Linux o/s. Hence portability problems will not arise.

Availability

This software will be available always.

Maintainability

The system called the ewheelz uses the 2-tier architecture. The 1st tier is the GUI, which is said to be front-end and the 2nd tier is the database, which uses MYSQL, which is the back-end.

The front-end can be run on different systems (clients). The database will be running at the server. Users access these forms by using the user-ids and the passwords.

Economic Feasibility

The computerized system takes care of the present existing system's data flow and procedures completely and should generate all the reports of the manual system besides a host of other management reports. It should be built as a web based application with separate web server and database server. This is required as the activities are spread throughout the organization customer wants a centralized database. Further some of the linked transactions take place in different locations Open source software like TOMCAT, JAVA, Mysql and Linux is used to minimize the cost for the Customer.

6.2.3 System Testing and Implementation:

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

Unit Testing:

Unit testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements.

Each module can be tested using the following two Strategies:

Black Box Testing:

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been used to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structure or external database access
- Performance errors
- Initialization and termination errors.

In this testing only the output is checked for correctness. The logical flow of the data is not checked.

White Box testing:

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It has been used to generate the test cases in the following cases:

- Guarantee that all independent paths have been executed.
- Execute all logical decisions on their true and false Sides.
- Execute all loops at their boundaries and within their operational bounds

Integrating Testing:

Integration testing ensures that software and subsystems work together as a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together.

System Testing:

Involves in-house testing of the entire system before delivery to the user. Its aim is to satisfy the user the system meets all requirements of the client's specifications.

Acceptance Testing:

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors.

Test Approach:

Testing can be done in two ways:

- Bottom up approach
- Top down approach

Bottom up Approach:

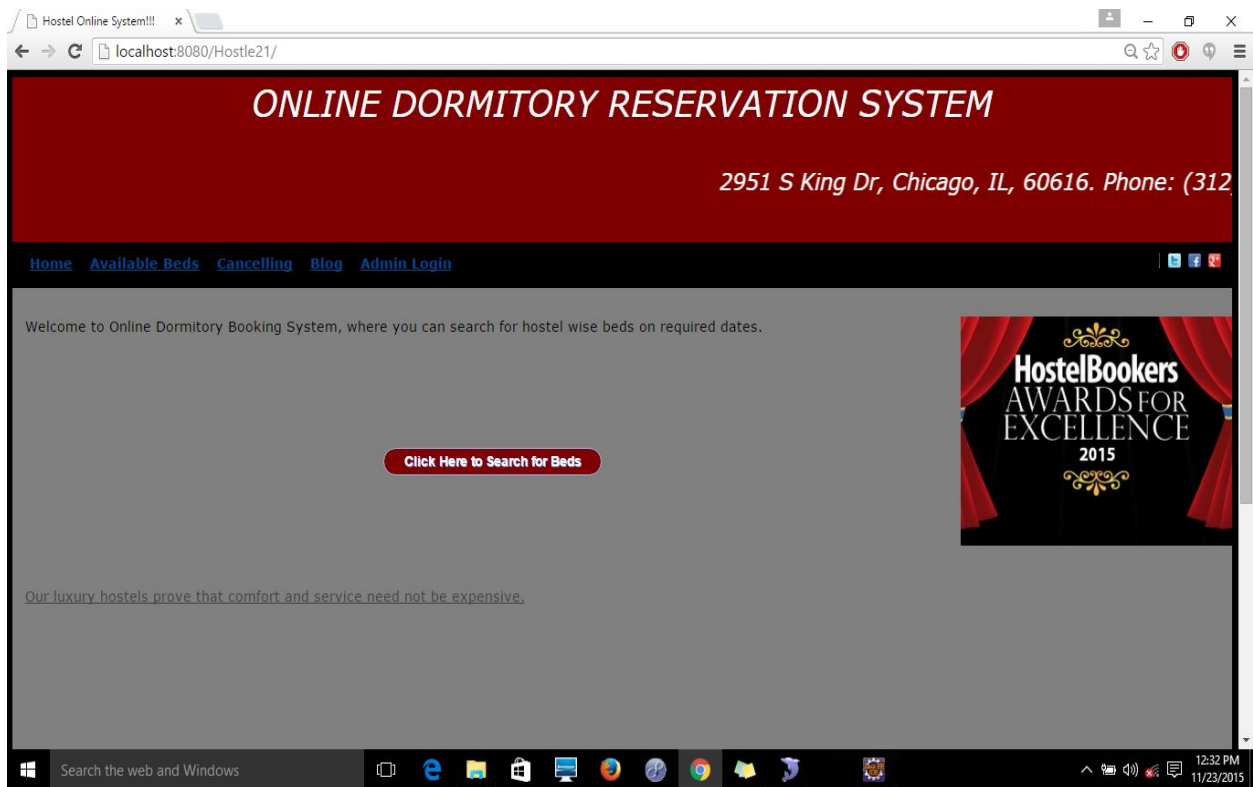
Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded within the larger system. When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules.

Top down approach:

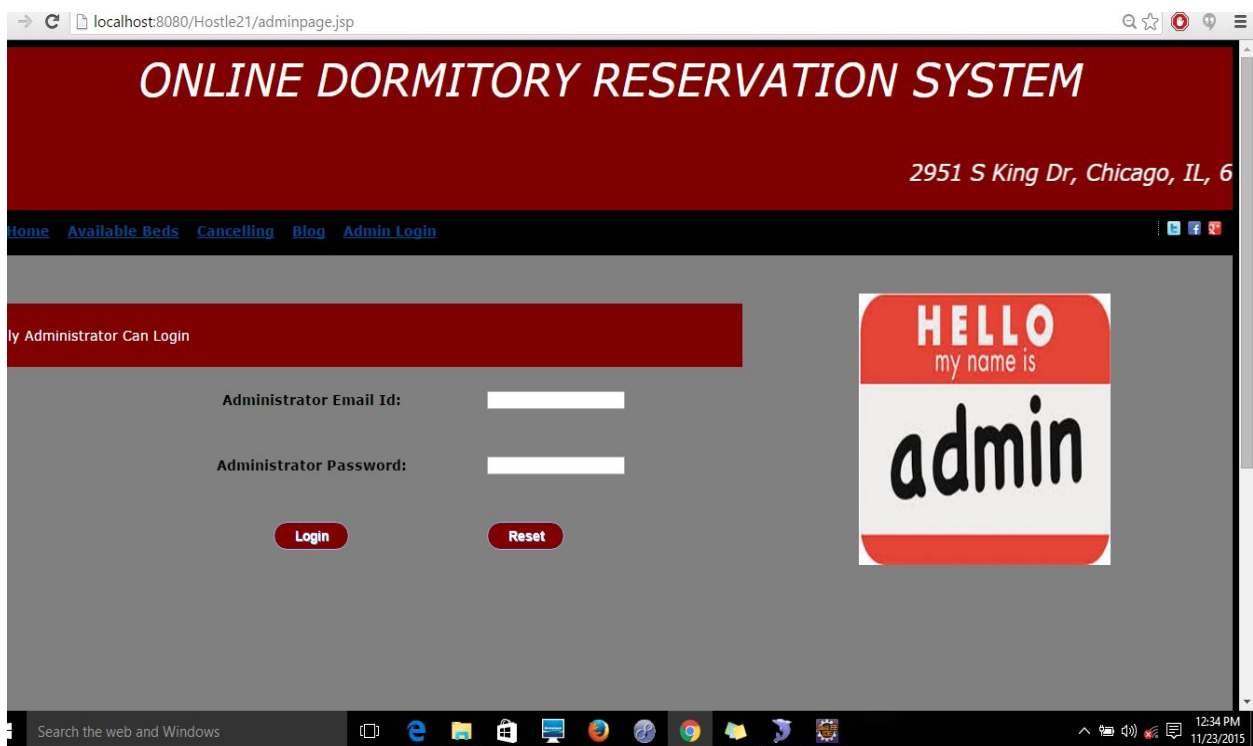
This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written. A stub is a module shell called by upper level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred. No attempt is made to verify the correctness of the lower level module.

6.3 OUTPUT SCREENS:

Home page:



Admin Login



Availability Page

Hostle # 1 Reservation x

localhost:8080/Hostle21/hostle1action.jsp

ONLINE DORMITORY RESERVATION SYSTEM

2951 S King Dr, Chicago, IL, 60616. Phone: (312)208-

[Home](#) [Available Beds](#) [Cancelling](#) [Blog](#) [Admin Login](#)

Available beds are :::

Booking Available Beds				
ROOM #	Bed	Dates Available	Cost	Select and Book
1	1	11-01-2015	\$12	<input type="checkbox"/>
1	2	11-01-2015	\$12	<input type="checkbox"/>
2	4	11-01-2015	\$12	<input type="checkbox"/>
3	5	11-01-2015	\$12	<input type="checkbox"/>
3	6	11-01-2015	\$12	<input type="checkbox"/>
1	7	11-01-2015	\$13	<input type="checkbox"/>
1	8	11-01-2015	\$13	<input type="checkbox"/>
4	9	11-01-2015	\$20	<input type="checkbox"/>
5	10	11-01-2015	\$20	<input type="checkbox"/>

[Book Now](#)

@2015 Dorms Online Booking System. All Rights Reserved

Search the web and Windows

12:33 PM 11/23/2015

Cancellation Page

Hostle # 1 Reservation x

localhost:8080/Hostle21/Cancelling.jsp

ONLINE DORMITORY RESERVATION SYSTEM

2951 S King Dr, Chicago, IL, 60616. Phone

[Home](#) [Available Beds](#) [Cancelling](#) [Blog](#) [Admin Login](#)

Cancelling booked beds

NOTE:

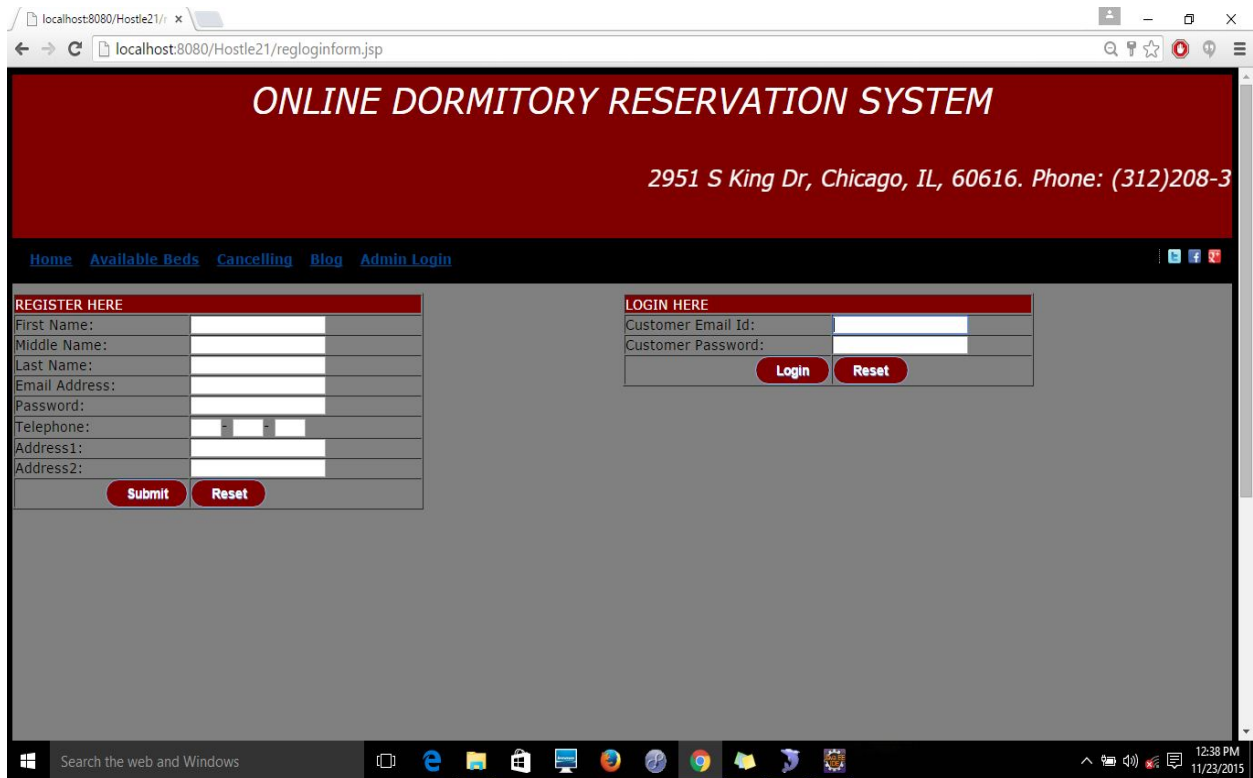
** If You Cancel the order the Cancellation Fee of \$5 will be Applicable to Your Refund

☐ Accepting the Cancellation Policy Terms and Conditions [Cancel Now](#)

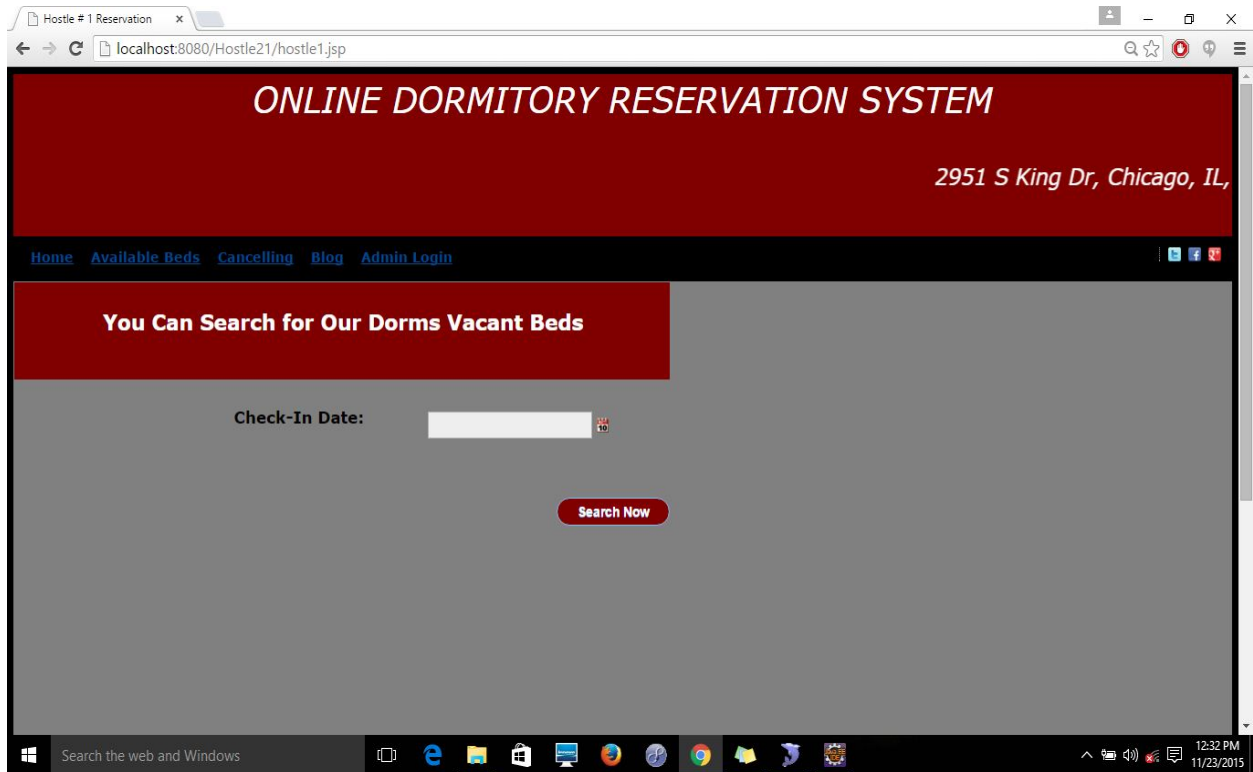
Search the web and Windows

12:33 PM 11/23/2015

User Login & Registration page



Check Vacant Beds



Overall booking Revenue Page

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/Hostle21/rangerevenue.jsp'. The page content is a table titled 'Overall Booking Revenue Details'. The table has four columns: 'ROOM #', 'Bed', 'Dates Booked', and 'Cost'. The first row contains the values '1', '1', '11-01-2015', and '\$12'. Below this row, there are two summary rows: 'Overall Booking Revenue:' with a value of '\$12', and 'Total Number of Bookings:' with a value of '1'. At the bottom of the table, there is a red text label 'Currently No Cancellation'. The browser's taskbar at the bottom shows the Windows logo, a search bar, and various application icons. The system clock in the bottom right corner indicates the time is 12:41 PM on 11/23/2015.

Overall Booking Revenue Details			
ROOM #	Bed	Dates Booked	Cost
1	1	11-01-2015	\$12
Overall Booking Revenue:			\$12
Total Number of Bookings:			1
Currently No Cancellation			

7. OPEN ISSUES

- Technical – Relating to a technological problem in the project.
- Business process – Relating to the project's design.
- Change management – Relating to business, customer, or environmental changes.
- Resource – Relating to equipment, material, or people problems.
- Third party – Relating to issues with vendors, suppliers, or another outside party.

8. ACKNOWLEDGEMENTS

We drive our great pleasure in expressing our sincere gratitude to advisor Dr.Soon Oak Park and Professor N.CHEN for their timely suggestions, which helped us to complete the project work successfully.

9 REFERENCES:

- Database System Concepts sixth edition by Abraham silbershatz, Henry F. Korth, S. Sudarshan.
- www.javatpoint.com.
- www.w3schools.com.
- <http://www.apachefriends.org>

10 APPENDICES