Hotel System:

Analysis, Design and Development

A dissertation submitted in partial fulfilment of

the requirements for the degree of

BSc (Hons) Computing

By

Kamran Khan

2012 - 2013

Declaration

No part of this project has been submitted in support of an application for any other degree or qualification at this or any other institute of learning. Apart from those parts of the project containing citations to the work of others, this project is my own unaided work.

Signed _____

Acknowledgements

I would like to thank my supervisors for aiding me during the process of this project.

Abstract

Conducting proper systems analysis, design and development techniques and methodologies is key, to producing any reliable and durable information system. Someone who wishes to produce such system should adhere to these principles. This report presents the various different techniques for each stage (Analysis, Design, Development, Testing and Evaluation) whilst developing the system, following the system life cycle stages and justifying the decisions made. The purpose of this project is to produce a fully functional web-based information system for a Hotel, which should then be tested and fully evaluated. The project is accompanied with this report, containing documentation and the processes undertaken to produce such system. The results produced from the evaluation, recognizes the system was well built and followed the common system methodologies. Additionally a brief discussion on improvements and further work which could be done is outlined.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

IS	Information System
IT	Information Technology
UML	Unified Modelling Language
SSADM	Structured Systems Analysis and Design
ERD	Entity Relationship Diagram
DFD	Diagram Flow Diagram
RDA	Relational Database Analysis
RAD	Rapid Application Development
OOP	Object Orientated Programming
HCI	Human Computer Interaction
SQL	Structured Query language
RDMS	Relational Database Management System
HTML	Hypertext Mark-up Language
CSS	Cascading Style Sheet
PHP	Hypertext Pre-processor
ASP	Active Server Pages
SMART	Specific, Measurable, Achievable, Relevant, Timely

1 Introduction

1.1 Project Background

In this technology age, there are many who can produce a system. However, producing a reliable, durable, maintainable and manageable system is a different story. Although many systems have many different concepts, the process of producing a hotel system in comparison to any other system is not all that different. However, having said that there are indeed different processes of producing such systems, this will be discussed later. Taking this into consideration, anyone looking to analyse, design and develop a system, should follow some common methodologies to achieve the above.

Producing a hotel system in particular can be challenging, there will be many aspects to take into consideration, not only on the development stage, but also on the analysis and design stages. Understanding what techniques to use for each stage is vital to producing a successful system.

1.2 Aims

The aim of the project is to produce a web-based management information system for a hotel. This will be accompanied with an academic report, outlining the process of producing such system, as well as explaining and justifying the techniques and methodologies used.

1.3 Objectives

In order to achieve the aim discussed above, a number of specific, measurable, achievable, realistic and timely objectives (SMART) (Doran, G. T., 1981) must be met. The objectives outlined below have been defined with the use and in accordance to the systems development lifecycle.

1.3.1 Research

1. Research Analysis, Design, Development, Testing and Evaluation Techniques.
2. Research current and previous Hotel Systems.

1.3.2 Analysis & Design

1. Gather and identify requirements.
2. Produce specification for functional and non-functional requirements.
3. Produce system Use case specification and diagrams from gathered requirements.
4. Produce system Data Flow model with explanation and justification.
5. Produce system Business Process/State Model with explanation and justification.
6. Produce system Data Dictionary with explanation and justification.
7. Produce system Relational Database Analysis with explanation and justification.
8. Produce system Class Model with explanation and justification.
9. Produce system Entity Relationship Model with explanation and justification.
10. Produce Wireframe designs of expected system with explanation and justification

1.3.3 Implementation

1. Develop front end to system.
2. Develop back end to system.
3. The system should allow users to login/register
4. The System should allow users to search for a room and reserve one.
5. The System should allow users to access their profile page.
6. The System should allow users to cancel reservations.
7. The System should allow users to update any details, including password.
8. The System should allow for an admin section
9. The System should allow admins to manage Rooms.
10. The system should allow admins to manage Reservations
11. The System should allow admins to manage Users

1.3.4 Testing

1. Create test strategies.
2. Identify test cases.
3. Carry out system testing.
4. Test system in terms of validation and compatibility

1.3.5 Evaluation

1. Produce Evaluation criteria.
2. Evaluate project against initial aims and objectives.
3. Evaluate project against initial requirements.
4. Evaluate project against user's acceptance.

1.3.6 Conclusion

1. Evaluate personal performance and improvements.

1.4 Report Structure

The main body of this report is broken down into six chapters. These chapters are listed below with a brief description.

Chapter2: Literature Review - An analysis and review of the academic knowledge in systems, analysis, design, development, testing and evaluation methodologies. As well as other literature relevant to this projects concept such as a review on current and previous hotel systems.

Chapter3: Analysis and Design - Describes the process of Analysis & Design according to the systems development lifecycle which involves: requirements gathering and identification, UML design and other techniques. This is all done, with the use of common standards used within the industry, which is documented with explanation and justification.

Chapter4: Implementation - Presents the implementation of the project, with the use of code fragments, accompanied with an explanation.

Chapter5: Testing – Will contain an overview of the testing techniques used for this project and show the process of testing and the results obtained.

Chapter6: Evaluation - An evaluation of the project against previously defined requirements and objectives, with the use of common evaluation techniques.

Chapter7: Conclusion - Overview of the entire project, outlining personal views with a discussion of defining the success of the project and elements for improvement. Finally, suggestions of further/improvements are outlined.

2 Literature Review

2.1 Introduction

This chapter contains the different academic knowledge in various fields, including but not limited to: Information Systems (IS) and System Analysis, Design and Development Methodologies. There exists a wide range of resources discussing different techniques and methodologies. The purpose of this chapter is to attain knowledge, required to produce a system, in-particular a hotel system. Furthermore, existing hotel systems are analysed and reviewed and finally various implementation technologies to produce an information system are outlined and discussed.

2.2 Industry Background

Demand for producing an information system for a Hotel or for multiple Hotels has been on the increase for many years. This is due to the convenience this service provides. Any online hotel today allows its users to search through available rooms by inputting some controlled data, such as: the desired check in and check out date. This simple system allows users to see the available rooms, allowing them reserve there desired room.

The online hotel system first made an appearance back in 1995. Where, Choice and Holiday in were the first to introduce online booking capability (American Hotel Lodging Association, n.d.). Since then more and more Hotel companies are looking to expand their presence online. The interest in this grew, because online information systems allow hotel companies to attract more customers and also display the services they offer. Potential customers could now finally have real time access to available rooms. Not only did an online information system allow hotels to broaden their scope of potential customers, but desktop information systems also allowed for a more efficient and effective management of their hotels. As the demand increased and as hotel companies became hotel chains, desktop management information systems allowed for the efficient management of the hotel.

Today many hotel information systems out there offer very efficient ways of booking a room. However, not many offer these solutions; offer an online management system and an effective booking system on a single website. Usually this would only cater for the big hotel chains. However,

with easy and effective information system developments, smaller companies should now be able to enjoy this.

2.2.1 What is an Information System?

To understand how to develop an information system, we should first fully understand what an information system is? What the role of an information system is and the important they are to businesses.

An Information system is any combination of information technology and people's activities that support operations, management and even decision making (SEI Report, Glossary, n.d.). However, today we know information systems to do much more than that. Broadly speaking, we can say it refers to the interaction between people, processes, data and technology. Furthermore, there are many different types of information systems such as: Desktop-Based Information Systems and Web-Based Information Systems. Although, there are others we will primarily be discussing these.

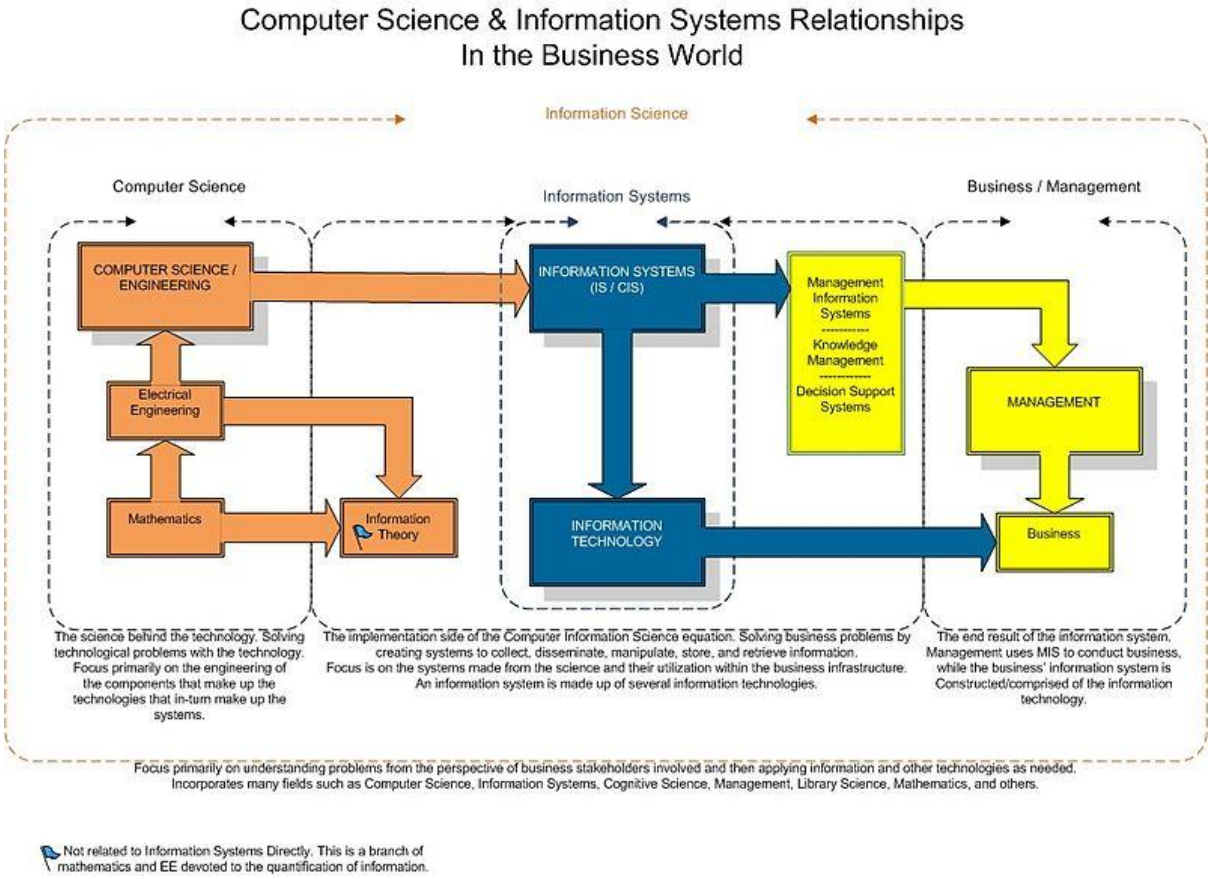


Figure 1 - Relationship between Information Systems and Business

Figure 1 as you can see shows the relationship between Computer science, information systems and business/management. An information system holds the position as a bridge between the science behind the technology and the end actions a business requires from an Information system.

2.2.2 The role of IS and the importance it has to Businesses

There are many different types of information systems out there. Therefore, naturally, information systems have numerous roles. The primary role of an IS, is to provide a business solutions to managing and/or online transacting and/or displaying organisational data and/or decision support making. Information systems today have huge impact on the operation, management and performance of businesses today. So much to the extent that businesses can't perform as a business without the use of an information system, we see this today in such industries as finance and banking. Beynon Davies states that "Businesses invest in information systems for one of three reasons: to do things more efficaciously, to be more efficient and to be more effective" (Beynon Davies, 2009). Each type of gain can lead to improved business performance. Therefore, making Information systems appealing to businesses and organisations alike.

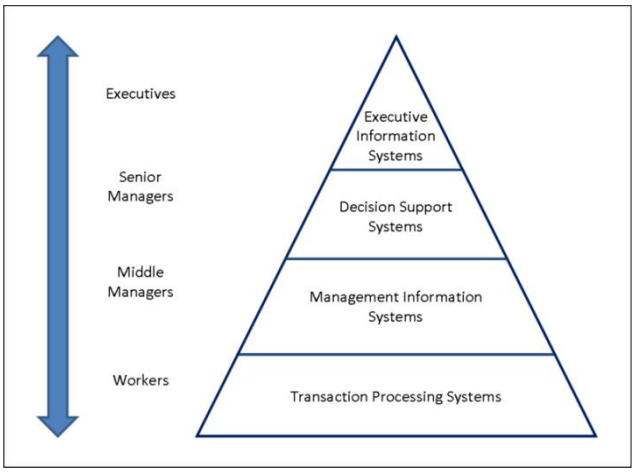


Figure 2 - Information Systems Roles

2.3 Systems Analysis & Design Methodologies and Techniques

2.3.1 Gathering and Documenting Requirements

Gathering and documenting requirements is one of the first steps to developing a system. In the development life cycle it comes under the Analysis stage. There are numerous industry known methods to gathering system requirements, this section of the literature review will discuss some of them as well as outlining the industry method of documenting these requirements.

Questionnaires are a very good tool for gathering requirements from stakeholders that are in a remote location. It's also a very good tool to gather input from a large source of people, thus, making this tool a more informal approach to requirements gathering. When it comes to systems development requirements gathering, questionnaires could be considered the first stage of requirements gathering. As they provide short answers, however, the need for more detailed answers when developing a system is seen as more important. Perhaps, the most common method in the industry today for requirements gathering is interviewing. Sitting down with the clients and asking them what they need is most popular as you can acquire the information you are looking for there and then. Furthermore, planning the interview ahead is ideal and in the industry it is generally recognized to ask open ended questions to engage the interviewee to uncover their requirements.

Once the requirements are gathered, usually these requirements will then be documented to be used in further stages of the system development life cycle. One approach to modelling these requirements is with the use of a Usecase diagram. Which is seen as the most commonly used technique in requirements gathering for IT systems (Debra Paul and James Cadle, 2010). We will take a detailed look at Usecase diagrams in the next part of this chapter. From the Usecases, Usecase specifications are produced which is just a detailed description of the Usecase diagram. This specification gets put into a requirements catalogue which consists of commonly used industry attributes including: Requirements ID, Name, Source, Priority, Type, Description, Pre and Post condition and finally Resolution (Debra Paul and James Cadle, 2010).

The process of requirements gathering and documentation outlined above is widely used in the industry. Although you may find some deviation, for the majority there is a common practise. Malcolm Eva in the BCS Business Analysis book states; that without sufficient time to understand and define the requirements' properly, the product that is delivered on time may not provide the

solution that the business thought was requested. Therefore applying the process of requirements engineering should help to rectify these problems (Malcolm Eva, 2010). From this we can note that the process of requirements gathering and documentation, although the least time is spent on it. In the development life cycle it is classed as one of the most important elements to building a successful system.

2.3.2 System Modelling - Logical Design & Physical Design

2.3.2.1 Logical Design

In the industry today there are four main system design/modelling practises which are widely used which will be discussed. These four include Dataflow Modelling, UML Modelling, E-R Modelling and finally Agile Modelling.

Data flow modelling is usually carried out under the Structured System Analysis & Design (SSADM) framework. This model is a technique which shows a graphical representation of the system intended for development or intended for analysis. It would generally show the dataflow within the system with the use of symbols including: The Source which is the entity or actor, the Process which is the function of the system and the Data store which is where the information being processed, which is stored usually a database and finally the Dataflow which shows the flow of information. Although it is a simple representation, it can be very confusing as there is not a set notation. Furthermore, it is a lengthy process where alterations are constantly made; therefore it is a slow process from start to finish.

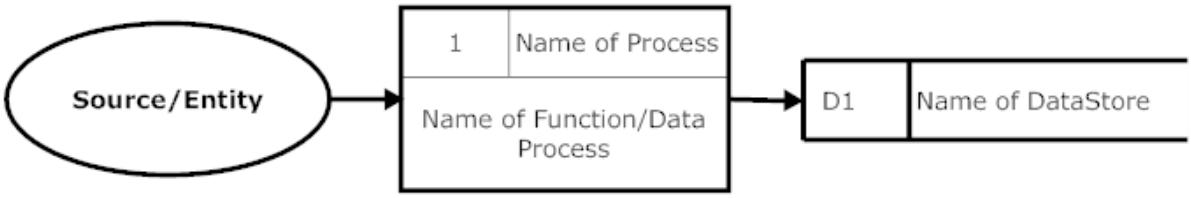


Figure 3 - Example Data Flow Model

UML modelling or Unified Modelling language is a very popular method for designing and modelling a system. The Unified Modelling Language (UML) is an Object-Oriented it is de facto standard for object oriented modelling language (Siau, K & Lee,L, 2004) . It is the language for specifying, visualising, constructing, and documenting the artefacts of software systems, as well as for business

modelling. There are 3 classification of this modelling methodology these are: Behaviour Diagrams, Interaction diagrams and finally Structure diagram.

UML Behaviour Diagrams are Usecase diagrams, although there are others, Usecase diagrams are the most popular UML behavioural diagram in the industry. Behaviour diagrams emphasize what must happen in the system being modelled. Since behaviour diagrams illustrate the behaviour of a system, they are used extensively to describe the functionality of software systems. Usecases diagrams generally consist of the actors which are the users of the system, the Usecases are the functions of the system and the associations are the relationship between the actors and the Usecases. As shown in Figure 4

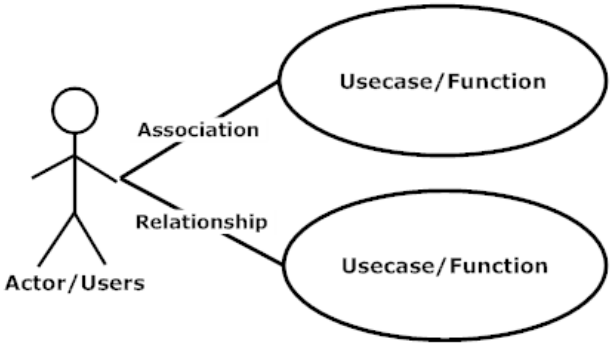


Figure 4 - Example Usecase Diagram

UML Interaction diagram is a subset of the Behavioural diagram. Interaction diagrams emphasize the flow of control and data among the things in the system being modelled. One of the popular diagrams in this classification is the Sequence Diagram. This diagram shows how objects communicate with each other in terms of a sequence of messages. Also indicates the lifespans of objects relative to those messages.

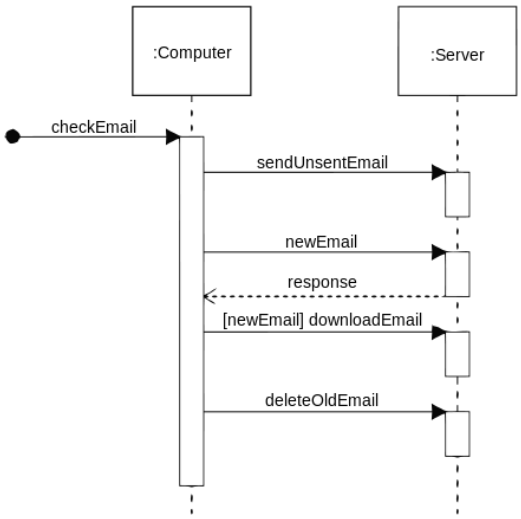


Figure 5 - Example Sequence Diagram

UML Structure diagrams emphasize the things that must be present in the system being modelled. Since structure diagrams represent the structure, they are used extensively in documenting the software architecture of software systems. Again the most widely used structure diagram is the Class Diagram. It describes the structure of a system by showing the system's classes, their attributes, and the relationships among the classes. As shown in Figure 6

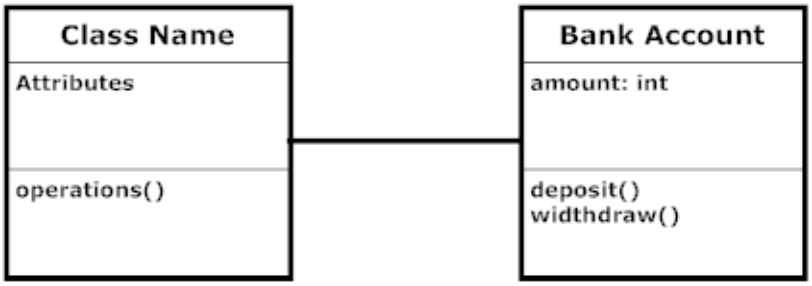


Figure 6 - Example Class Diagram

Unlike dataflow modelling, UML harbours a variety of different diagrams which all have different benefits. It is no shock that UML is the most commonly used modelling methodology in systems design and development today.

ER Modelling is very limited and is an abstract way to describe a system which primarily uses a database. ER model generally consists of entities, relationships, and attributes. The entity-relationship model adopts the more natural view that the real world consists of entities and relationships. It incorporates some of the important semantic information about the real world (Peter Chen, 1976). However, it does have its limitations unlike UML modelling it does not show the behaviourally and interaction of the system but only the structural view.

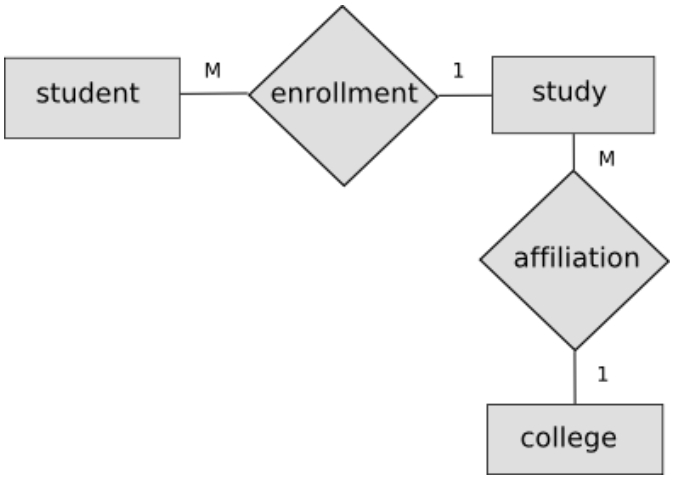


Figure 7 - E-R Model Chens Notation

Agile Modelling is a fairly new modelling methodology. Agile Modelling (AM) is a practice-based methodology for effective modelling and documentation of software-based systems. At a high level AM is a collection of best practices, depicted in the pattern language. At a more detailed level AM is a collection of values, principles, and practices for modelling software that can be applied on a software development project in an effective and light-weight manner. (Scott W. Ambler, 2005). This suggests that agile modelling can be a very flexible process, where the need to constantly make changes is appropriate with this type of methodology.

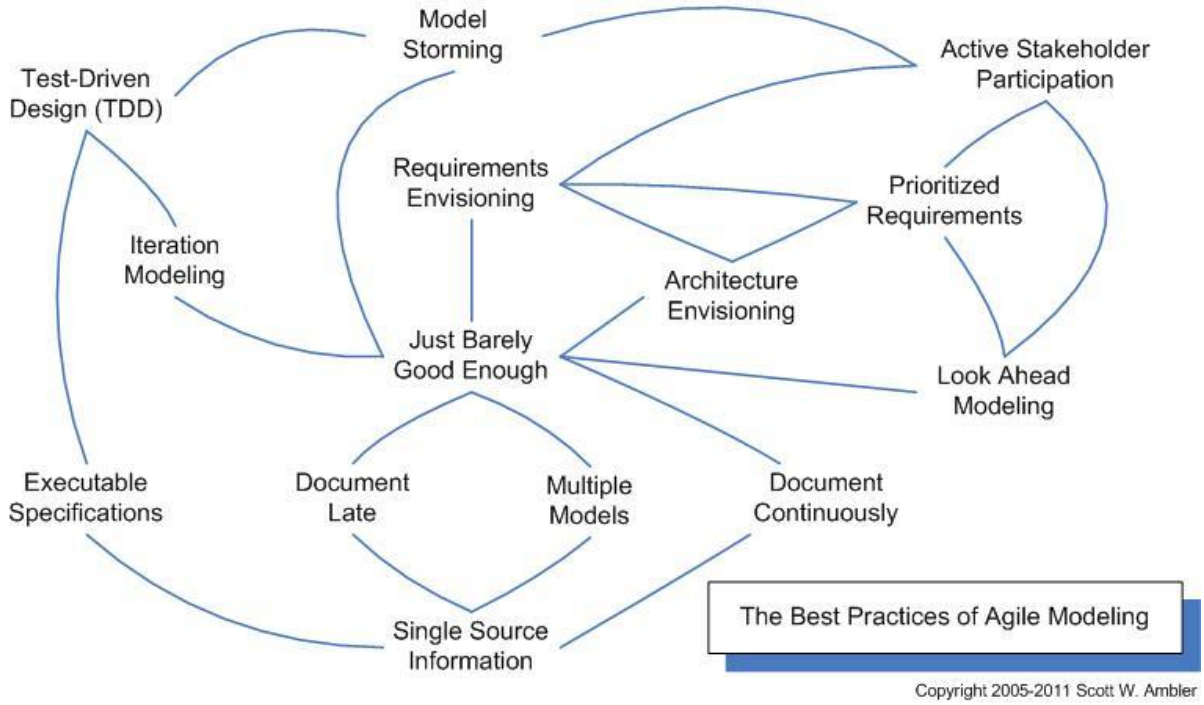


Figure 8 - Best Practises of Agile Modelling

From the literature above we can clearly identify that the different modelling methodologies all having their own benefits and drawbacks. However, you could say that UML modelling and notation is by far the best approach. As, like the name suggests, it unifies a lot of the modelling techniques and diagrams into a single notation practise. Although the agile modelling approach is very useful in that it is flexible, the modelling approach really depends upon the approach you take to developing the system. This will be further discussed in a later section of the literature review.

2.3.2.2 Physical Design

Now that we have discussed different techniques of logically designing a system, let's look at some methods for the physical design of a system which is sometimes known as prototype design.

There are a few physical design techniques used over the years when developing information systems. Some of them are described below.

Storyboarding – Is probably the simplest method of prototyping. Storyboarding is a graphical representation of the intended systems appearance and interface. Storyboarding tends to leave out any system functionality, although, it may indicate the low-level functionality of the specific graphics in the storyboard. Storyboarding allows quick evaluation from clients, to determine whether this design is in line with the clients/users requirements. Benefit of storyboarding is that it is not costly; it does the job intended and requires very limited time. (Dix, A; Finlay, J; D.Abowd,G; Beale,R, 2004)

Limited Functionality Prototype – A Limited functionality of system is built to demonstrate the system. With the use of graphics and animation, simulation of the functionality is produced; however, the high-level programming is not there. As this prototyping mimics the functionality, evaluation, like storyboarding can be quick. However, there may be some cost incurred such as the time, as this type of prototyping could consume a lot of time, but may seem little initially. (Dix, A; Finlay, J; D.Abowd,G; Beale,R, 2004).

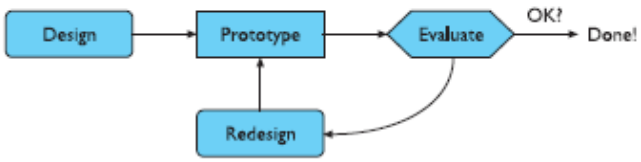


Figure 9 - Prototype Design

Iterative Design/Prototyping – Is a gradual design and development of the system to ensure all requirements are defined and met. A small section of the system is built, new requirements are defined and the system is then changed to meet these new requirements, after each iteration the design is improved, there is an increase in functionally and finally the end product will reflect the intention. The iterative design tends to come to an end when no more requirements and improvements are identified. (Dix, A; Finlay, J; D.Abowd,G; Beale,R, 2004).

There are significant drawbacks to iterative and prototype designing. Such as, the initial design decision made at the beginning of the process could be wrong. At a later stage in the prototype phase it may be too difficult to fix these initial design problems. Time can also be a very big concern.

2.3.3 Human Computer Interaction

Many information systems today are developed and built with poor human computer interaction taken into account. So what exactly is HCI? It is the process by which systems should be designed with an understanding of the end user and specific tasks in mind. So that the end product has been designed for the purpose of the intended audience.

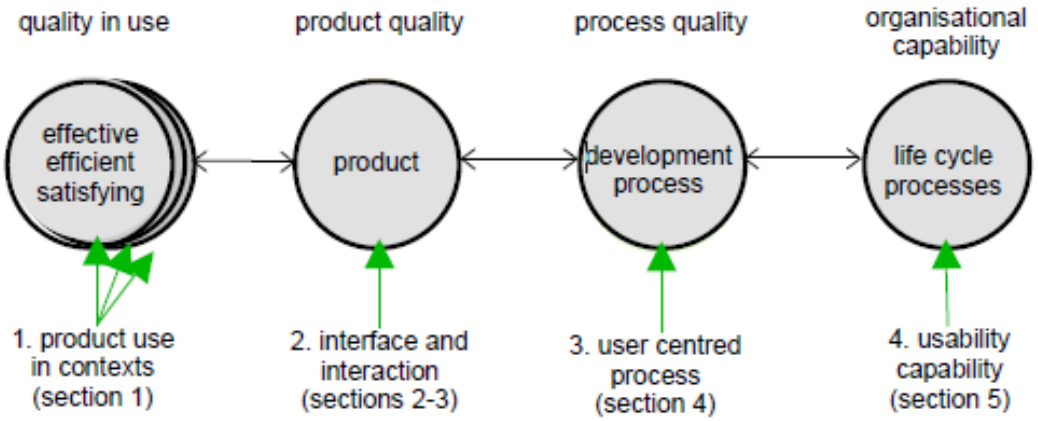


Figure 10 - HCI Categories of Standard

The international HCI standards state standards related to usability can be categorised as primarily concerned with:

1. The use of the product (effectiveness, efficiency and satisfaction in a particular context of use).
2. The user interface and interaction.
3. The process used to develop the product.
4. The capability of an organisation to apply user centred design.

Figure 10 shows the logical relationships: the objective is for the product to be effective, efficient and satisfying when used in the intended contexts. A prerequisite for this is an appropriate interface and interaction. This requires a user centred design process, which to be achieved consistently requires an organisational capability to support user centred design. [Citation Needed]

As the development of this project is primarily focused on a web-based information system, it is therefore, important for us to look at the foundation guidelines for producing a web based information system in line with the HCI common practises. Initial users will base their assumption of the web-based information system by what they initially see, in this case it would be the homepage

for the website. Therefore, it is important to know the guidelines for a homepage design. (Nielsen, J & Tahir, M., 2001) (Krug, S, 2006). Some of these guidelines are listed below, w hich are most relevant to the project being undertaken.

1. Company name and logo should be displayed in a noticeable position.
2. User should identify what the site has to offer from initial look.
3. High Priority Functions should be easily identified and usable.
4. Corporate Information should be easy accessible
5. Navigation should be easy to find and use
6. Registration and Sign in System
7. Unique Homepage for attraction
8. Appropriate Download Time.

These guidelines are appropriate for a web-based information system and fulfil the 3 ‘use’ words to describe a successful system in terms of HCI. Useful, usable and used. HCI is definitely something to think about when designing and developing an information system.

2.4 System Development Methodologies and Techniques

Just like the different analysis and design methodologies, you can find in the industry today. There are indeed numerous different system development methodologies and techniques used. Many of the earliest information systems, developed before the 1960's were developed without the use of many of the systems development methodologies we have today (Avison, D & Fitzgerald,G, 2003). Because of this many of the systems developed at that time, did not fulfil the requirements intended and often were development later than the intended time frame. Due to this an introduction of systems development methodologies were needed. We define this as a collection of procedures, techniques, tools and documentation aids which help developers to implement systems (Avison, D & Fitzgerald,G, 2003). We find today there are many different systems development methodologies which suit many development styles. This part of the literature will discuss the different development methodologies and determine their benefits and drawbacks in comparison to one another and the project being undertaken.

2.4.1 Waterfall Model

The waterfall models first formal description is often cited as a 1970 article by Winston W. Royce. The waterfall model to software development is a sequential design and development process. The main idea of this model is take the large system development tasks and break them down into smaller and more manageable sub tasks. The stages of the waterfall model are requirements, design, implementation, testing and maintenance. The model is referred to as a waterfall as each progress/stage is seen as flowing steadily downwards. Essentially, the waterfall model requires the fulfilment and completion of one stage before continuing to the next. As shown in Figure 11

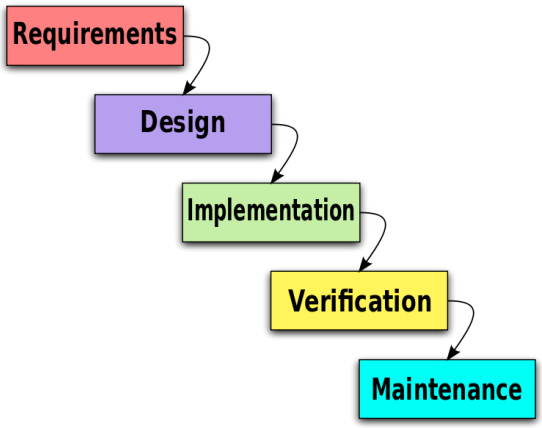


Figure 11 - Stages of the Waterfall Model

The waterfall model has some good supporting arguments, such as: each stage is well defined, dividing complex task of producing a system into small sub tasks, helps for better organisation. Another supporting argument made by McConnell is that a bug found in the early stages such as the requirements specification or design is cheaper in money, effort and time to fix than the same bug found later on in the process (McConnell, 1996). However, there are some criticisms concerning this type of development. Dawson sheds light on one of the main problems of this model. He states as we move through the stages of the process, the problem we are trying to solve is not static and moves on. Thus by the time we complete our analysis and design and building of the software system, the problem we pretend our user no longer solves the problem they currently have – i.e. the software has already become obsolete. (Dawson, C, 2009). However, Dawson does go on to state that waterfall model is appropriate for projects where the length of the project is no more than one year. He also states projects which are clearly understood and the initial requirements are captured accurately and specification and designs match closely with what is needed can succeed with the use of this development approach.

2.4.2 SSADM

SSADM or Structured System Analysis and Design methodology is probably one most traditional systems development methodologies. SSADM concentrates primarily on the analysis and design stage of the waterfall model of the systems development lifecycle (HNComputing, 2007). It has many of the strengths and weaknesses as the waterfall model as that is what it is based upon. However, this type of development concentrates on the thorough studying of the business from different perspectives, which could yield greater development results in line with the business/co-operate objective.

2.4.3 Rapid Application Development

Rapid Application Development (RAD) is software development approach that uses minimal planning and documentation in favour of rapid prototyping of the system. The lack of pre-planning allows for the software to be coded much faster, which makes it easier to change the requirements. There are four stages to RAD: Requirements planning phase, user design phase, construction phase and cutover phase. Benefits of RAD are that it promotes strong collaborative atmosphere and dynamic gathering of requirements. Business owners actively participate in the prototyping, test cases and performing unit testing. However, there is no degree of centralised project management.

2.4.4 Agile Development

Agile development is seen a contrast to the traditional waterfall development. Whereas the waterfall model is static and flows downwards, the agile approach is based on iterative and incremental development. After each iteration it is intend to have a working software program, not without its faults though. Agile approach is based upon 4 principles (AgileManifesto, 2001).

- **Individuals and Interactions** over processes and tools.
- **Working Software** over comprehensive documentation.
- **Customer collaboration** over contract negotiation.
- **Responding to change** over following a plan.

That is, while there is value in the items on the right, we value the items on the left more. (AgileManifesto, 2001).

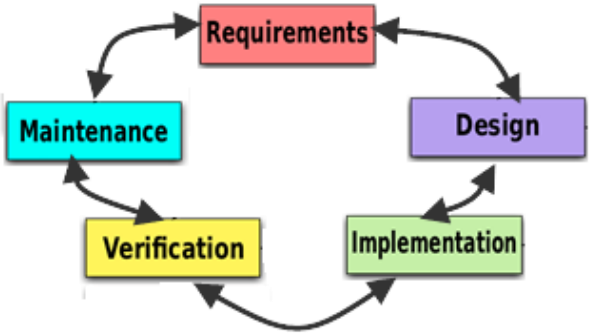


Figure 12 - Agile Model

Some of the arguments for agile model: Is Agile approach is well suited to projects which have unclear or rapidly changing requirements (Dawson, C, 2009). Some criticism of this approach is that agile methodology is inefficient in larger organisations and certain types of projects. Agile methods seem best for developmental of non-sequential projects. Another drawback is the project can easily get taken off track if the customer representative is not clear what final outcome that they want. However, many other benefits are; it has an adaptive team which is able to respond to the changing requirements, Face to face communication and continuous inputs from customer representative leaves no space for guesswork, the end result is the high quality software in least possible time duration and satisfied customer.

2.5 Analysis of Existing Hotel Systems

There are currently two different types of Hotel Information systems that we can classify into groups. Group 1, are the hotel systems which are predominantly desktop-based information systems and Group 2, are the hotel systems which are web-based information systems. This section of the literature review, will review the different classifications of Hotel Systems in terms of analysing the different design and development methodologies and technologies used and finally assessing the success of such information systems based on industry standard assessment criteria and criteria specific to development of Hotel Information Systems.

There are a number of ways to assess an information system, different types of information systems may be assessed in different ways. However, for this purpose we will look at general assessment factors. An assessment of an information system can determine many factors including but not limited to: Appearance, Performance, Functionality and Information.

2.5.1 Group 1: Desktop-Based Hotel System

Desktop-based information systems are software solutions which are generally kept on a local machine or a local server with ease of access to all the information. In the hotel industry today, you can find some examples of these desktop software solutions albeit, very few examples. Two of the desktop-based information systems, being analysed and evaluated here you can find in Appendix 4.1 for KW Hotel and Appendix 4.2 for ASI Front Desk.

As you can see from these two hotel systems, they both have very common functionality between them. These common functionalities include: allowing staff and or managers to manage rooms, customers and reservation in terms of adding, updating and deleting records. As well, as having functionality for extra features such as reporting. Therefore, we can identify some type of resemblance and default functions that a hotel system specifically a management system should have when it comes to managing a hotel.

Furthermore, we can identify from the two desktop software's that they were both designed and developed using similar technologies. From the layout and structure we can identify that they resemble the Microsoft layout and structure, suggesting they have been developing using Microsoft Visual Studio as opposed to another technology. This also suggests to us that they have used one of

Microsoft's programming languages, either Visual Basic C# or C++. This is generally seen as a good practise to some, as the users seem more familiar with programs developed in such a manner, as they have used other systems with similar layout, such as Microsoft's popular Office for example.

In terms of appearance desktop-based information systems are not known to be aesthetically pleasing, this is because such types of information systems concentrate more on the functionality of the intended system, rather than the look of the system. However, like mentioned previously, they do factor in the feel and use of the system when trying to figure out the best design and development methodologies and techniques to use, to make it easy to use.

One noticeable aspect after using one of the above desktop-based hotel systems, is that there is no indication of a back-end database. A back-end database can be very good for many reasons, including but not limited to storing data and backing up data. This can be very inefficient at times.

To conclude the desktop-based information systems although the most oldest type of hotel system, there are some simple flaws mentioned above. However, we can conclude that functionally of desktop-hotel systems are similar. Also, they are similar with web-based hotel systems which will be looked at further in Group 2. Finally, one can see that this type of information system has been addressed to one particular stakeholder of a hotel, which would be the managers, as the tool only caters for them providing them information. However, it does not cater for the customers of a hotel.

2.5.2 Group 2: Web-Based Hotel System

Web-based information systems are systems which use a website as a front end interface. Web-based information systems are stored and hosted on a server, therefore accessible anywhere, unlike desktop-based information systems we have found in group 1. Some of the hotels discussed below you can find the appendices, for booking.com refers to appendix 4.3 and for laterooms.com refer to appendix 4.4.

The web-based information systems we can find in Group 2 are different in ways in comparison to Group 1, whereas in Group 1, hotel systems concentrate on the management of a hotel, group 2 hotel systems concentrate on the reservation systems of a hotel, providing a broad search function for many hotels.

In terms of functionality again we can see a similarity between all of the 2 systems in group 2 e.g. they all provide functionality for a login and registration system, this is one of the key technical/functional requirements which will need to be integrated into any web-based hotel system. The next common functionality they have between the hotel systems Is search function, the search functions is an important technical requirement for a hotel system as it allows the user to query the system based on information entered to yield hotels according to the desired requirement. Furthermore, they all use similar data fields for this query such as a “check in” and “check out” field. Again suggesting these systems like group 1 all have common functionally.

In terms of the technology used to develop these systems we can only speculate. Assuming they are a website they have used HTML and CSS to design their website. However when it comes to programming and scripting, laterooms.com has used asp.net and we cannot determine what booking.com has implemented. However, we can also determine that they have a backend database to store information and retrieve results from the search query as well as storing data about customers and rooms. As these types of web-based information systems, are known as Dynamic Websites.

For appearance u can see that all 3 systems in group 2 are aseptically pleasing, as we know this type of system is developed for the stakeholder of the customer, thus this web-based information system must be attractive to draw attention. Some of the look and layout of these websites have been already discussed in the HCI section of the literature.

In conclusion, both Group 1 and Group 2 hotel systems have many things in common and elements which are distinct from one another. The reason for this is that both hotel systems are developed for different stakeholders, thus providing a different outlook and required information, each holding their own benefits and drawbacks. However, it has determined that the systems common to each other in both groups hold some very similar functionality, indicating that all hotel systems must have the above functionality to be considered either a web-based hotel system or a desktop-based hotel system.

2.6 Analysis of System Development Technology

2.6.1 Relational Database Management Systems

A database is just a structural set of data. A database allows for storing managing and the retrieval of data. SQL is a tool for organising, managing and retrieving data stored by a computer database (Groff,J R & Weinberb, P N, 2002) .SQL is essentially a computer programming language that you use to interact with the database. In order to control the database, you need to utilise a database management system. So why use a database? Using a database in-conjunction with an information system can be very useful. It allows you to display information stored on the database which users can then interact with e.g. storing information about hotel rooms in a database. Then displaying this information which allows users to interact with the information carrying out functions such as add, update and delete. In the RDMS market today there are 3 main competitors which will be reviewed.

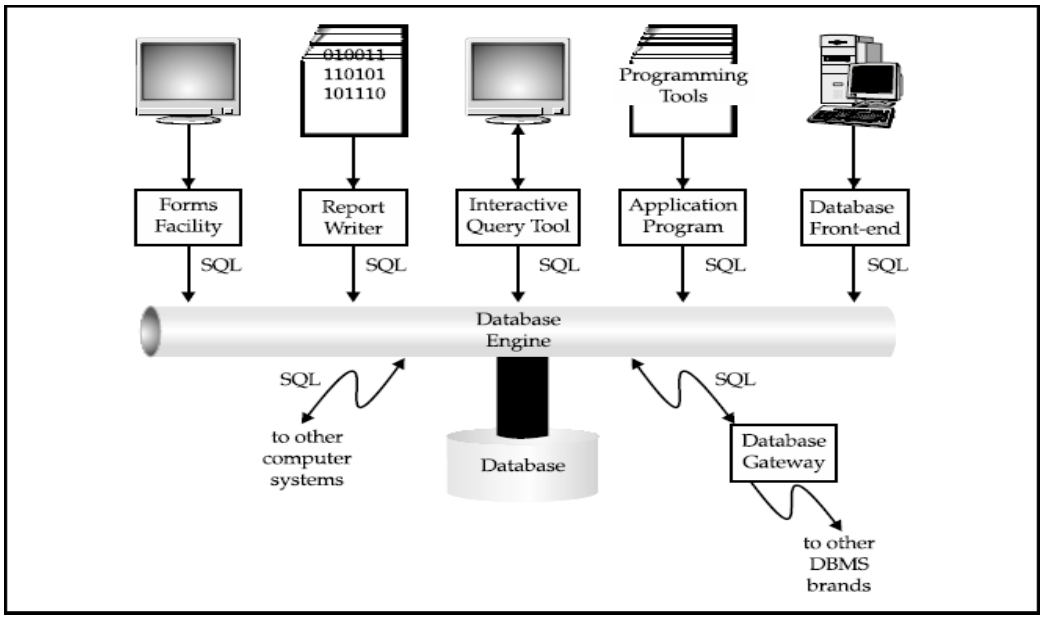


Figure 13 - Components of Database Management System (Groff,J R & Weinberb, P N, 2002)

2.6.1.1 Oracle

Oracle's current RDMS version is Oracle Database 11G, with various editions. Oracle began in 1979, as the first commercially available relational database management system (Kenneth Hess, 2010). Oracle is free to develop, deploy and distribute. Oracle also supports many internal programs within the database system such as data mining.

2.6.1.2 MySQL

MySQL is an open source database. It started off as a very niche database management system for many developers, but now has become a major contender in the enterprise database market (Kenneth Hess, 2010). MySQL is now owned by Oracle Cooperation since 2008 (Kenneth Hess, 2010). MySQL now powers many commercial websites. Some of the benefits of MySQL apart from being open source is that it is free, easy to use and runs on any operating system platform.

2.6.1.3 SQL Server

Generally companies using Microsoft OS platforms and frameworks will go for Microsoft's SQL Server, they provide a free version of their RDMS. Many small and big enterprises today are using Microsoft's SQL server, as it is very easy to use and integrate with other Microsoft technologies.

Although Oracle and MySQL are good choices to go with, for such a small project like this. SQL server will be ideal as it provides many of the benefits Oracle and MySQL are offering and much more, in terms of cost again it is free like the previous two RDMS's mentioned. Primarily it will work in harmony with other Microsoft technologies/languages be used for this project and it is the most secure RDMS. As you can see from figure 2.5.1.3.1 Microsoft SQL Server is a very popular RDMS and more people are planning to choose it over Oracles RDMS, in the future.

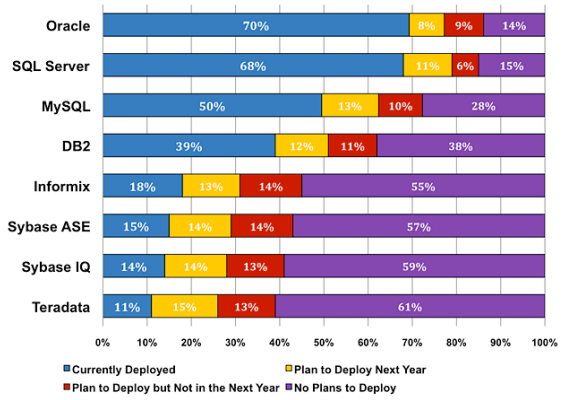


Figure 14 - Database Management System Plans

2.6.2 Client Side Mark-up Language and Style Sheet Language

2.6.2.1 HTML & CSS

Client side scripting is generally about programming the behaviour of the Browser. HTML and CSS is the standard client side mark-up language and style sheet used in the industry today when developing web-based information system. As there is no real alternative, there is no other contender to debate over. HTML is the mark-up language used for displaying information on a web-Brower. CSS is the style sheet language used for describing the look and formatting of a document written in mark-up language HTML. Both languages are crucial to developing any web-site. Most importantly you do not any need special software to write these languages in or to deploy, they will work in notepad, but most commonly used in notepad++ and Adobe's Dreamweaver. Furthermore, as the intention for this project is to use Microsoft's Technology. Microsoft's Integrated Developing Environment (IDE) integrates both of these languages very well.

2.6.3 Server Side Scripting Programming Language

Server side scripting unlike client side scripting it is generally about programming the behaviour of the server. This is why it is called server-side scripting or server scripting. Server side scripting can have many benefits such as dynamically editing, changing or adding any content to a web page. Additionally, helps with responding to user's queries or data submitted. Futhermore, it aids with accessing any data or database and return the results to a web browser, customising a web page to make it more useful for individuals and finally provide security. (w3schools, N/A). There are 2 main server side scripting languages we will take a look at.

2.6.3.1 PHP

PHP is a free open source server side scripting language, owned by PHP licence. It is the most widely used scripting languages because firstly as mentioned it is free, it is easy to learn and has an extensive API documentation and built-in functionality for many common tasks. PHP can also be used on most OS platforms.

2.6.3.2 ASP.Net

ASP.NET is Microsoft's counterpart to PHP, again ASP.NET is a free server side scripting languages licenced by Microsoft. Just like PHP it is relatively easy to learn and costs nothing to run on a server.

As mentioned previously for this project, we will be using many other Microsoft Technologies and Languages to develop the web-based information system. Thus, it would only seem appropriate to use Microsoft's ASP.Net Language. As it provides many of the capabilities PHP does but most importantly it goes hand in hand with Microsoft's SQL Server and Microsoft's IDE.

2.6.4 Web Server

This leads on to the web server, as previously mentioned Microsoft Technologies will be used, it would only make sense to use Microsoft's Internet Information Service (IIS) as opposed to any other web server service.

2.7 System Testing Techniques

2.7.1 Why We Test Systems

Testing system is an important factor in developing any systems. One could say there are many reasons why developers test their systems. At the end of it, it comes down to quality management and reducing cost caused by errors. Therefore, there are many testing techniques today, some more appropriate and valid for certain systems than others. In this section of the literature review, 3 common testing techniques will be outlined which are relevant to developing a web-based information system.

2.7.2 Black Box Testing

Black box testing is the testing of business functions. This approach would usually involve testing all aspects and combinations of end-user actions, such as login/search/registration/management. With black box testing, the tester assumes no knowledge of the code, however, they intend to simulate the target audience experience. Like other testing techniques, black box testing does not need to be carried out only when the entire system is completed, but instead, it can be done gradually through the whole systems development lifecycle (Microsoft Corporation, 2005).

2.7.3 White Box Testing

White Box testing is the opposite of black box testing, in white box testing the tester would generally create some test cases by looking at the code to detect any possible scenarios which would cause failure or error. Suitable input data for testing would be needed. Also to note with white box testing, test plans need to be produced first and should be kept updated before starting white box testing and only after a stable build of the code is available, should it be performed.

2.7.4 Website Validation and Compatibility

As web-based information systems become more and more popular, the need for testing the validity and compatibility of the code produced for online use has become more crucial. Having websites work in multiple browsers has also has been advantageous, as not all your users will be using the

same software. Validation of code using standards set such as w3 validator should be a beneficial tool to use.

2.8 Summary

To summarise in the literature review presented, research was undertaken to describe the foundations of developing an information system. Firstly, some industry background was outlined, along with the purpose of what an information system is and its benefits to businesses. Furthermore, different analysis and design methodologies were analysed, presenting the benefits and drawbacks for each and relating its suitability back to the initial aims of the project. Moreover, system development methodologies were discussed, again, presenting the positive and negatives of each and how each would cater to the development of certain information systems. Additionally, industry hotel information systems were analysed and evaluated, to determine their strengths and weaknesses and to see what could be gained e.g. what were the industry standard. Finally, system development methodologies were looked at to see what they could offer when building dynamic web-based information systems. Also, testing technique were briefly touched upon

To conclude, the research done in this literature has presented much useful and vital information needed in order to build a robust and efficient information system for a hotel. This chapter of the dissertation has achieved the objective of: researching analysis, design and development techniques along with current system. As the knowledge has been acquired on how to produce such system. The Analysis, designing, development and testing of the product can begin.

3 Systems Analysis and Design

3.1 Introduction

The purpose of this chapter is to carry out the analysis and design techniques and methodologies, researched in the literature review. The aim at the end of this chapter is to develop the system based on the techniques used in this chapter. This chapter will briefly touch upon the requirements gathering techniques used to gather the specifics of the system. Then, the requirements specification for this web-based information system will be outlined. Furthermore, a requirements catalogue will be produced documenting each of the specification aims, with the use of Usecase specifications and diagrams, accompanied with state diagrams to provide additional information on the requirement function dynamics. Moreover, a dataflow model will be produced to represent the flow of information needed. Furthermore, the design stage will be implemented; this will cover many of the essential modelling techniques used to develop a system, such as a class model and an entity relationship model. And finally, mock-up wireframe prototype designs will be produced for the intended system.

3.2 Requirements Gathering

The gatherings of the requirements were based upon the review of existing and previous web-based information system which was specific to hotel systems.

3.3 Requirements Catalogue

This part of the analysis and design chapter, will catalogue the requirements obtained in the previous section (3.2). The Requirements will be catalogued into a table, the process of cataloguing each requirement will be split up into the Users/Customers section and the Admin section. Each table will consist of standard Usecase specification elements you would find used in the industry today, refer to (Malcolm Eva, 2010) chapter on requirements gathering. These elements include, Identifier also the code name of the requirement, the function which is what the requirement has to do, the source which is where the requirement was obtained, the priority level of the requirement, the type of requirement it is functional/non-functional, the actor is who the requirement is intended for, the pre-requisite is what needs to be done before the requirements can take place and the post condition of the requirement is what happens after the function is carried out. Additionally, a brief requirement description, acceptance criteria and resolution element will be present. Each set of tables will be accompanied with a Usecase diagram, to diagrammatically view the function and a state diagram, again to diagrammatically view the dynamic /change state of each function.

3.3.1 Users

3.3.1.1 Registration

Requirements Catalogue – User Registration		
Date: -	**Version: 1.0**	**Status: -**

Requirement Identifier	G-CFN01Regv01
Requirement Function	To Allow Users To Register
Source	-
Priority	M
Type Of Requirement	Functional
Actor	A User
Pre-Condition	User must fill in registration details, and pass validation checks.
Post-Condition	User must confirm registration, through clicking on link which has been sent to his email. They then become a registered user. User Profile data but be auto generated
Requirement Description	The system should all users to register
Acceptance Criteria	None
Resolution	To be implemented in the development stage of this project

Table 1 - Requirement User Registration Specification

1. A Users Goes Onto The website
2. Navigates to Registration Page
3. Fills In Registration Details
4. Details Not Accepted Return Back To [2]
5. Details Accepted
6. Registration Email Gets Sent
7. User Confirms Email
8. User is now a Registered User.

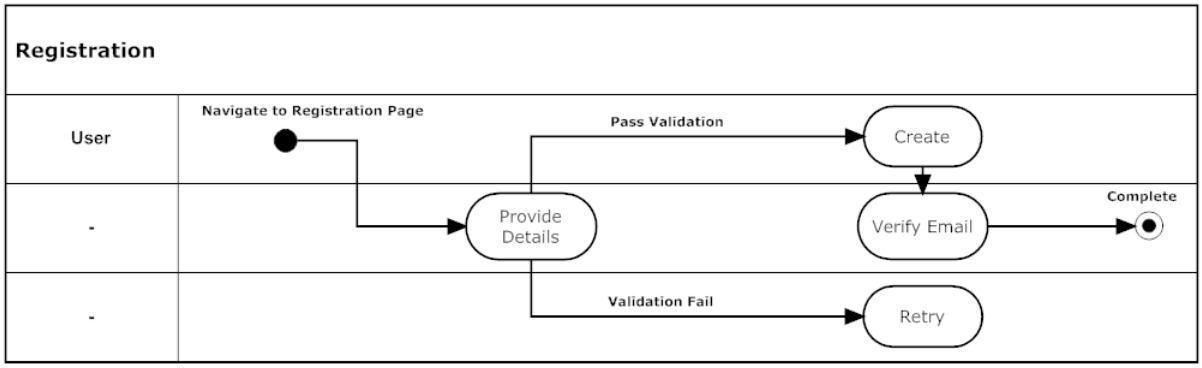


Figure 15 - Requirement User Registration State Diagram

3.3.1.2 Login Function

Requirements Catalogue – User Login		
Date: -	**Version: 1.0**	**Status: -**

Requirement Identifier	G-CFN02Loginv01
Requirement Function	To Allow Users To Register
Source	-
Priority	M
Type Of Requirement	Functional
Actor	A Registered User
Pre-Condition	A user must be registered before they can login
Post-Condition	none
Requirement Description	The system should allow all registered users to login
Acceptance Criteria	None
Resolution	To be implemented in the development stage of this project

Table 2 - Requirement User Login Specification

1. User Goes Onto The Website
2. User Navigates To Login Page
3. User Must Be Registered - User Refer to Function [3.3.1]
4. Registered User Forgets Password - Refer to Function [3.3.3]
5. Registered User Fills in Details
6. Details Are Not Accepted
7. Registered User Returns Back To [2]
8. Details Accepted
9. Registered User is now Logged In.

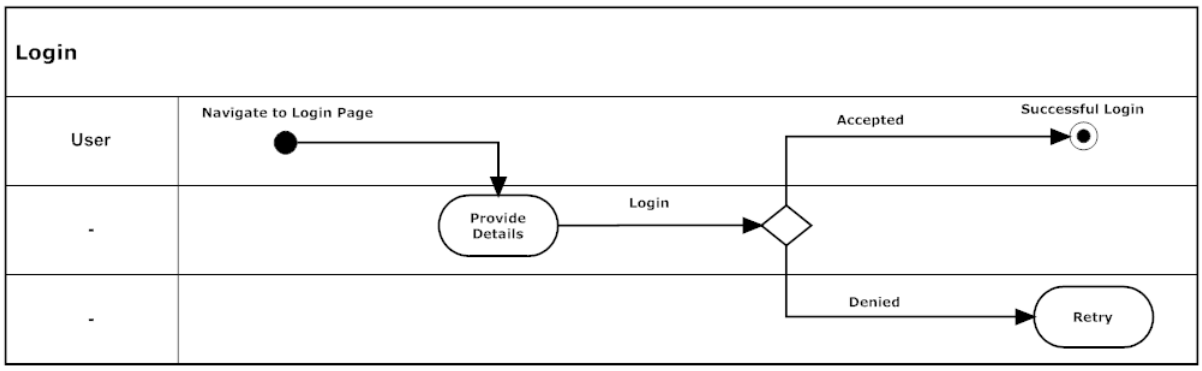


Figure 16 - Requirement Login State Diagram

3.3.1.3 Forgot Password

Requirements Catalogue – Forgot Password		
Date: -	**Version: 1.0**	**Status: -**
Requirement Identifier	G-CFN03Passwordv01	
Requirement Function	To Allow Registered Users to Retrieve password	
Source	-	
Priority	M	
Type Of Requirement	Functional	
Actor	A Registered User	
Pre-Condition	A user must be registered before they can retrieve password	
Post-Condition	Email Gets Sent With Password	
Requirement Description	The system should allow all registered users to retrieve there password	
Acceptance Criteria	None	
Resolution	To be implemented in the development stage of this project	

Table 3 - Requirement User Forgot Password Specification

1. Registered User Forgets Password
2. Registered User Provides Username
3. Password Gets Sent To Email Associated With Username
4. Registered User Retrieves there Password

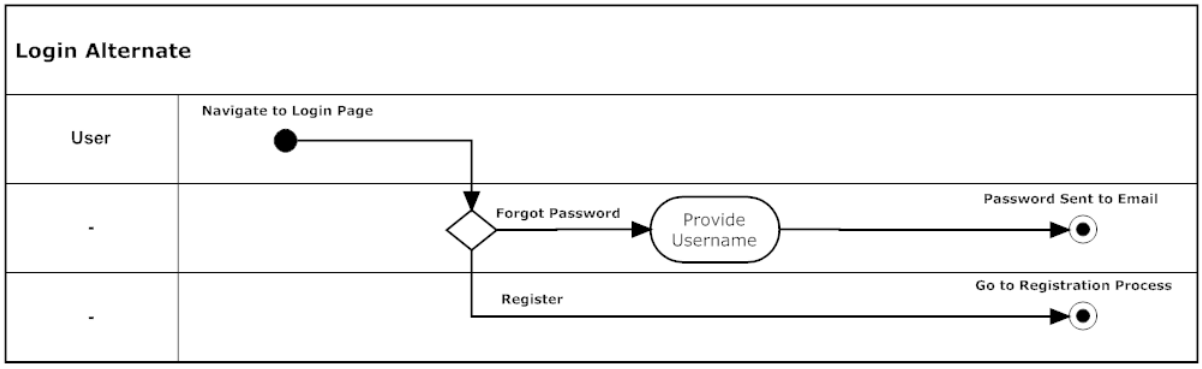


Figure 17 - Requirement Forgot Password State Diagram

3.3.1.4 Change Password

Requirements Catalogue – User Change Password		
Date: -	**Version: 1.0**	**Status: -**

Requirement Identifier	G-CFN04CPv01
Requirement Function	To Allow Registered Users To Change Password
Source	-
Priority	M
Type Of Requirement	Functional
Actor	A Registered User
Pre-Condition	A user must be registered and logged in before they can change password
Post-Condition	Confirm Email Sent
Requirement Description	The system should allow all registered users to change password
Acceptance Criteria	None
Resolution	To be implemented in the development stage of this project

Table 4 – Requirement User Change Password Specification

1. Registered User Logs In
2. Navigates To Profile Page
3. Navigates To Change Password Page
4. Provides Username, Original Password and New Password
5. Confirmation of Password Email gets sent.

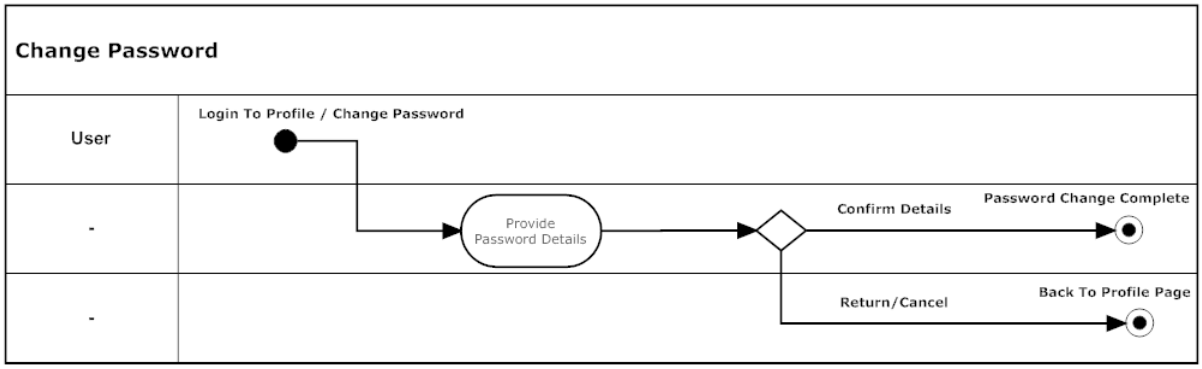


Figure 18 - Requirement Change Password State Diagram

3.3.1.5 Update Profile

Requirements Catalogue – Update Profile Details		
Date: -	**Version: 1.0**	**Status: -**
Requirement Identifier	F-CFN05UpdPfeDtlsv01	
Requirement Function	To update profile details	
Source	-	
Priority	M	
Type Of Requirement	Functional	
Actor	Registered User	
Pre-Condition	Registered User Must be Logged In	
Post-Condition	Profile details have been updated to database	
Requirement Description	The system should allow users to update their profile details	
Acceptance Criteria	Only registered users are given the ability to update their profile details.	
Resolution	To be implemented in the development stage of this project	

Table 5 – Requirement User Update Profile Specification

1. Registered User Logs In
2. Navigates To Profile Page
3. Navigates to Update Profile Page
4. Fills In Profile Details
5. Registered User Confirms Changes
6. Details are updated into database

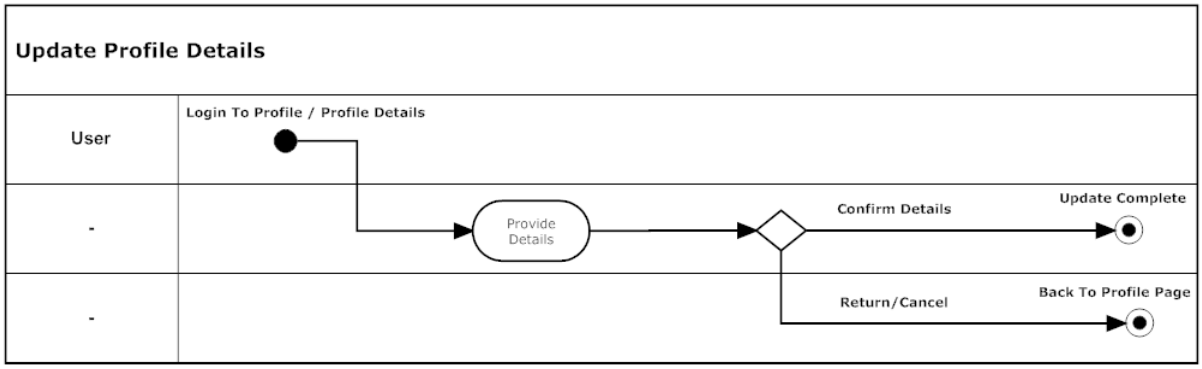


Figure 19 - Requirement Update Profile Details State Diagram

3.3.1.6 Book Reservation

Requirements Catalogue – Book a Reservation		
Date: -	**Version: 1.0**	**Status: -**

Requirement Identifier	F-CFN06BookRNv01
Requirement Function	To book a room for reservation
Source	-
Priority	M
Type Of Requirement	Functional
Actor	Registered User
Pre-Condition	Must be loged in, must find room to reserve first.
Post-Condition	Reservation details logged into database.
Requirement Description	The system should allow only registered users to book a reservation.
Acceptance Criteria	Only users who have registered can book a room for reservation and the room must available for reservation.
Resolution	To be implemented in the development stage of this project

Table 6 – Requirement User Book Reservation Specification

1. User Navigates to Search Page
2. User Inputs Desired Dates
3. User Finds Desired Room
4. User Reserves Room – Denied Must be Registered User
5. Registered User Reserves Room
6. Registered User Confirms Reservation

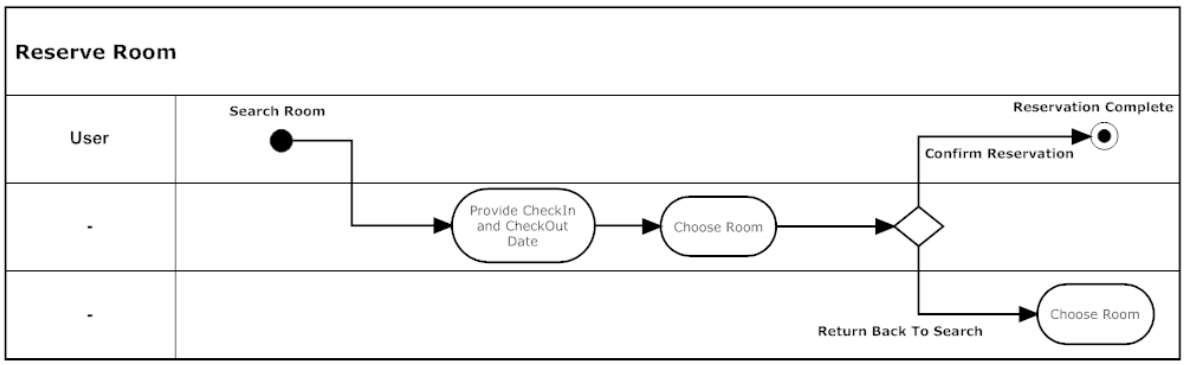


Figure 20 - Requirement User Book Reservation State Diagram

3.3.1.7 Cancel Reservation

Requirements Catalogue – Cancel a Reservation		
Date: -	**Version: 1.0**	**Status: -**

Requirement Identifier	F-CFN07CancelRNv01
Requirement Function	To cancel a reservation
Source	*
Priority	M
Type Of Requirement	Functional
Actor	Registered User
Pre-Condition	Must have a reservation to cancel, must first find reservation.
Post-Condition	Reservation removed from database.
Requirement Description	The system should all users to cancel a reservation the user is associated with
Acceptance Criteria	Only users who are associated with a particular reservation can cancel the reservation.
Resolution	To be implemented in the development stage of this project

Table 7 – Requirement User Cancel Reservation Specification

1. Registered User Logs In
2. Registered User Navigates to Profile Page
3. Registered User Navigates To Reservation Management Page
4. Registered User Selects Reservation For Cancel
5. Registered User Confirms Cancellation

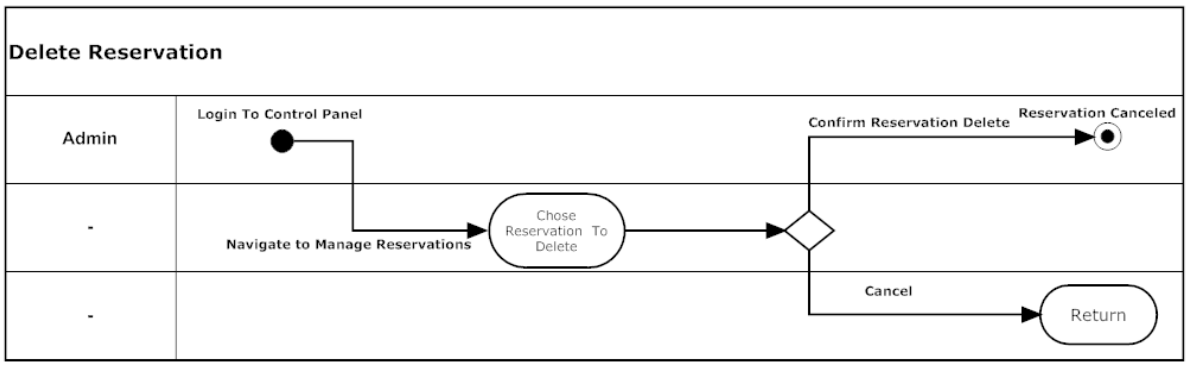


Figure 21 - Requirement User Cancel Reservation State Diagram

3.3.1.8 User Functions Usecase Diagram

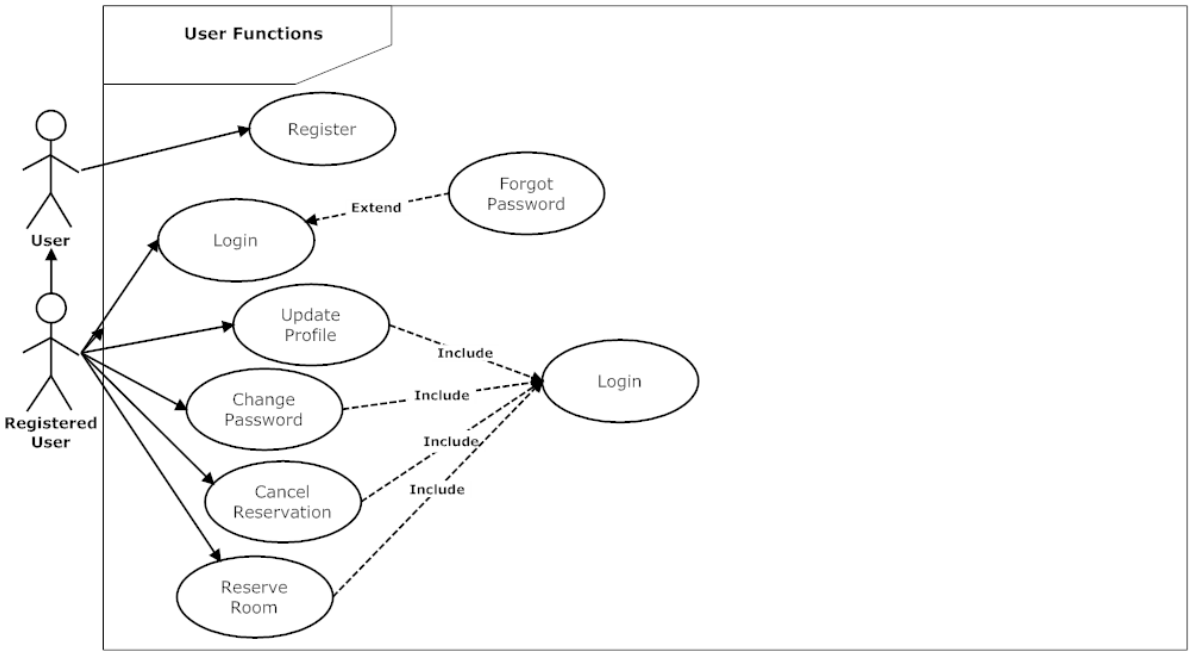


Figure 22 - User Functions Use Case Diagram

3.3.2 Admin

3.3.2.1 Room Management

Requirements Catalogue – Insert		
Date: -	**Version: 1.0**	**Status: -**
Requirement Identifier	F-RM01Insertv01	
Requirement Function	To Insert Rooms to the System	
Source	-	
Priority	M	
Type Of Requirement	Functional	
Actor	Staff/Admin	
Pre-Condition	Knowledge of Room Details Required	
Post-Condition	Room is now added.	
Requirement Description	The system should allow users (staff) members to add rooms to the system at any time.	
Acceptance Criteria	Only users with authorised access are allowed add rooms to the hotel.	
Resolution	To be implemented in the development stage of this project	

Table 8 – Requirement Room Insert Specification

1. Admin Logs In
2. Navigates To Control Panel
3. Navigates To Room Management – Insert Room
4. Inserts New Room Details
5. Confirms Insert
6. Details Updated To Database

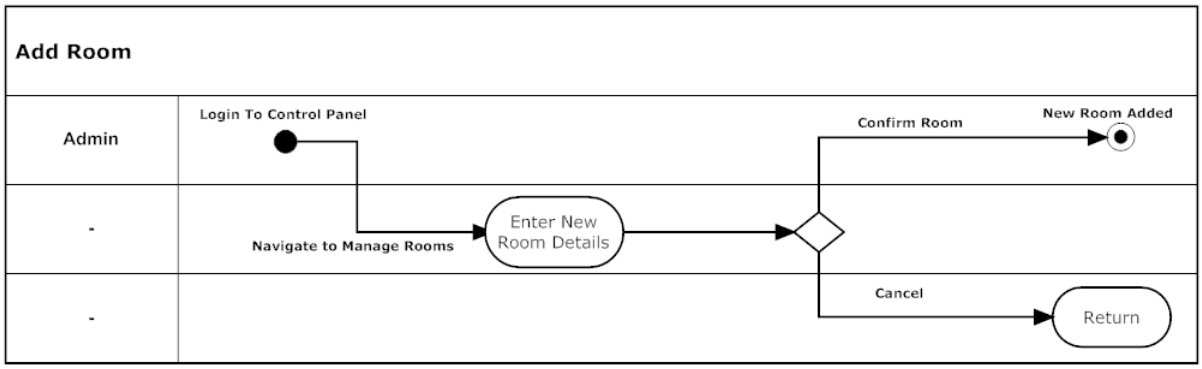


Figure 23 - Requirement Room Management Add Room State Diagram

Requirements Catalogue - Update		
Date: -	**Version:** 1.0	**Status:** -

Requirement Identifier	F-RM02Updatev01
Requirement Function	To Update Room Details
Source	-
Priority	M
Type Of Requirement	Functional
Actor	Staff/Admin
Pre-Condition	Must first select which room to update.
Post-Condition	Room is updated, with new details.
Requirement Description	The system should allow user (staff) members to update room details of the system at any time.
Acceptance Criteria	Only users with authorised access are allowed to update room details
Resolution	To be implemented in the development stage of this project

Table 9 – Requirement Room Update Specification

7. Admin Logs In
8. Navigates To Control Panel
9. Navigates To Room Management – Update Room
10. Updated New Room Details
11. Confirms Update
12. Details Updated To Database

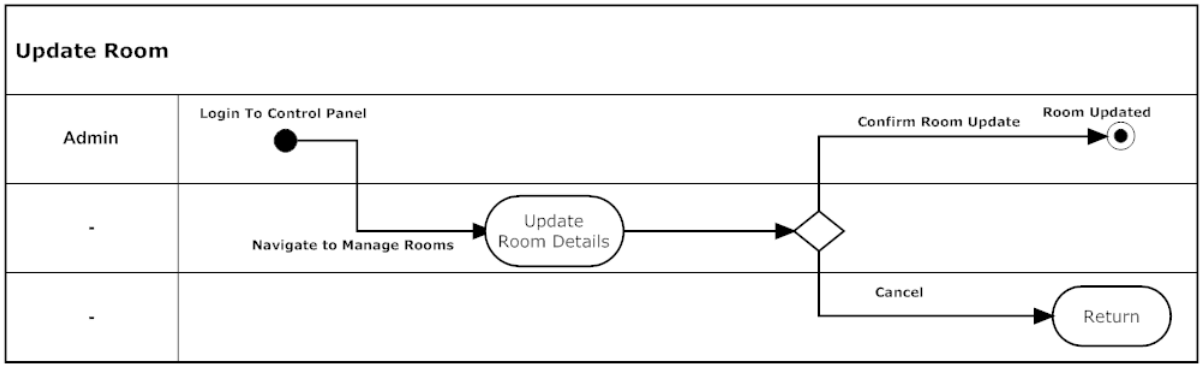


Figure 24 - Requirement Room Management Update Room State Diagram

Requirements Catalogue – Delete		
Date: -	**Version:** 1.0	**Status**: -

Requirement Identifier	F-RM03Deletev01
Requirement Function	To Delete Rooms
Source	-
Priority	M
Type Of Requirement	Functional
Actor	Staff/Admin
Pre-Condition	Find room to remove.
Post-Condition	Room has been removed from database.
Requirement Description	The system should allow user(staff) members to delete rooms from the system
Acceptance Criteria	Only users with authorised access are allowed to delete rooms from the system.
Resolution	To be implemented in the development stage of this project

Table 10 - Requirement Room Delete Specification

1. Admin Logs In
2. Navigates To Control Panel
3. Navigates To Room Management – Delete Room
4. Finds Room To Delete
5. Confirms Delete
6. Deletion Updated Into Database

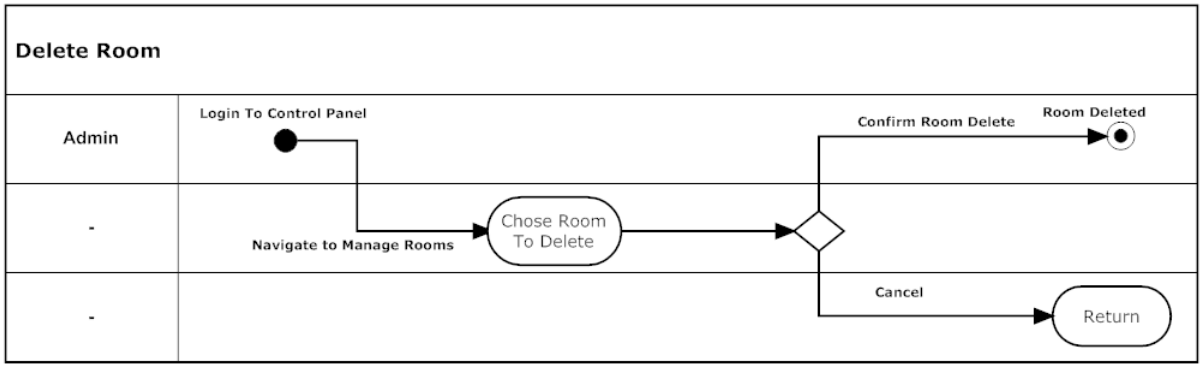


Figure 25 - Requirement Room Management Delete Room State Diagram

3.3.2.2 Reservation Management

Requirements Catalogue - Delete		
Date: -	**Version: 1.0**	**Status: -**

Requirement Identifier	F-RN03Insertv01
Requirement Function	To Cancel/Delete A Reservation
Source	-
Priority	M
Type Of Requirement	Functional
Actor	Staff/Admin
Pre-Condition	Find Reservation to remove.
Post-Condition	Reservation has been removed from database.
Requirement Description	The system should allow users (staff) members to cancel/delete a reservation for a room.
Acceptance Criteria	Only users with authorised access are allowed to delete reservation from the system.
Resolution	To be implemented in the development stage of this project

Table 11 - Requirement Reservation Update Specification

1. Admin Logs In
2. Navigates To Control Panel
3. Navigates To Reservation Management – Delete Reservation
4. Finds Reservation To Delete
5. Confirms Delete
6. Deletion Updated Into Database

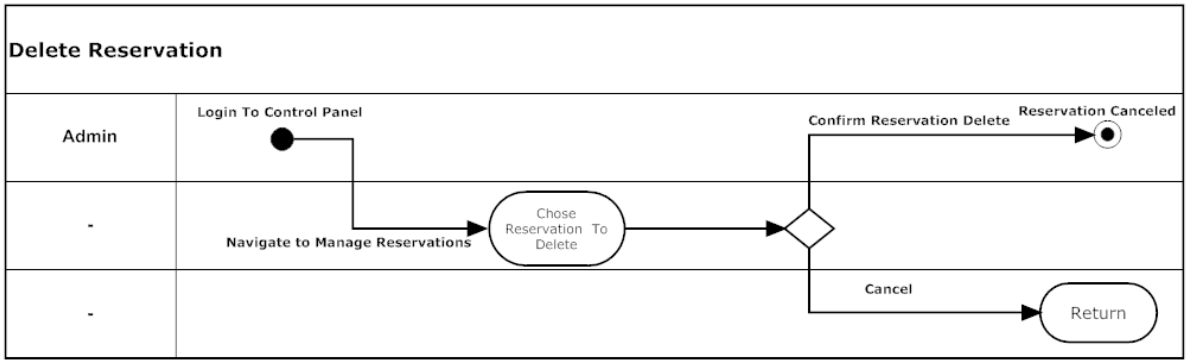


Figure 26 - Requirement Reservation Management Delete Reservation State Diagram

3.3.2.3 Report Management

Requirements Catalogue – Produce Customer Report		
Date: -	**Version: 1.0**	**Status: -**
Requirement Identifier	F-RP01CstRPv01	
Requirement Function	To Produce Customer Reports	
Source	-	
Priority	S	
Type Of Requirement	Functional	
Actor	Staff/Admin	
Pre-Condition	Select Desired Information	
Post-Condition	Report Produced.	
Requirement Description	The system should all staff members to produce reports of the customers	
Acceptance Criteria	Only users who have authorised can produce reports.	
Resolution	To be implemented in the development stage of this project	

Table 12 – Requirement Report Customer Specification

1. Admin Logs In
2. Navigates to Control Panel
3. Selects range of customers
4. Builds Report

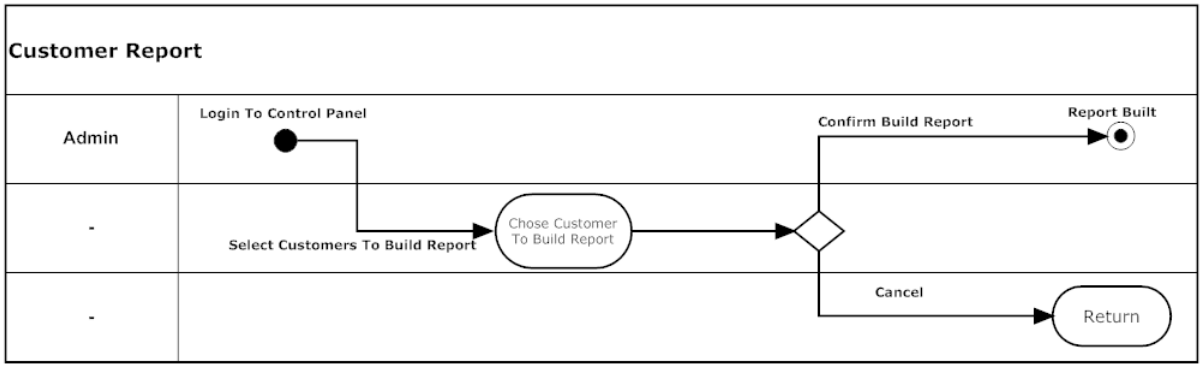


Figure 27 - Requirement Report Customer State Diagram

Requirements Catalogue – Produce Reservation Report		
Date: -	**Version: 1.0**	**Status: -**
Requirement Identifier	F-RP01CstRPv01	
Requirement Function	To Produce Reservation Reports	
Source	-	
Priority	S	
Type Of Requirement	Functional	
Actor	Staff/Admin	
Pre-Condition	Select Desired Information	
Post-Condition	Report Produced.	
Requirement Description	The system should all staff members to produce reports of the reservation	
Acceptance Criteria	Only users who have authorised can produce reports.	
Resolution	To be implemented in the development stage of this project	

Table 13 - Requirement Report Reservation Specification

1. Admin Logs In
2. Navigates to Control Panel
3. Selects range of dates of reservation
4. Builds Report

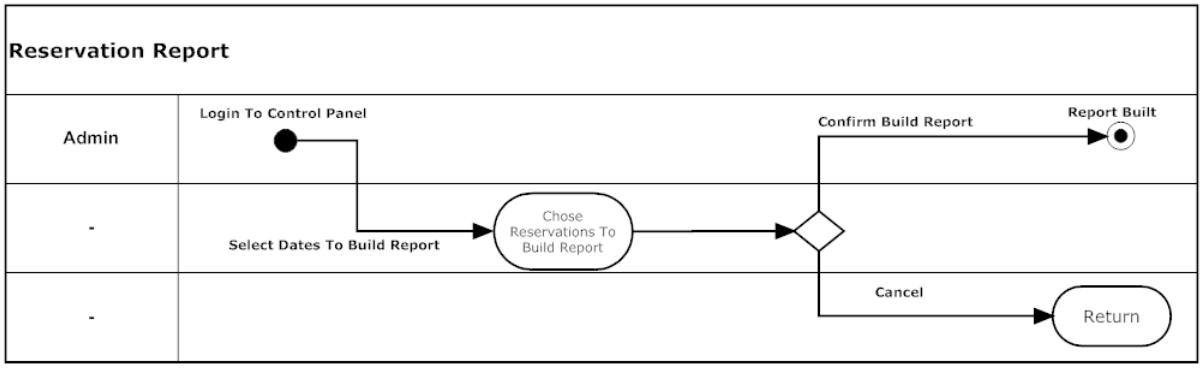


Figure 28 - Requirement Report Reservation State Diagram

3.3.2.4 Admin Use Case Diagram

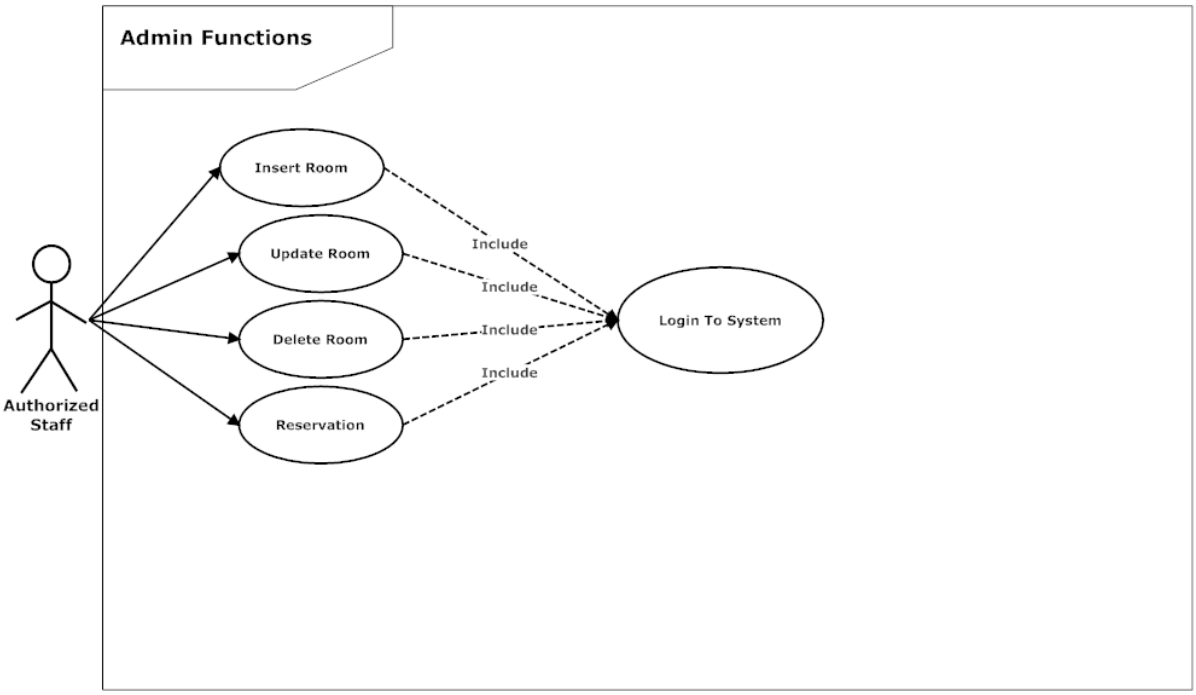


Figure 29 - Admin Functions Use Case Diagram

3.3.3 System Use Case Diagram

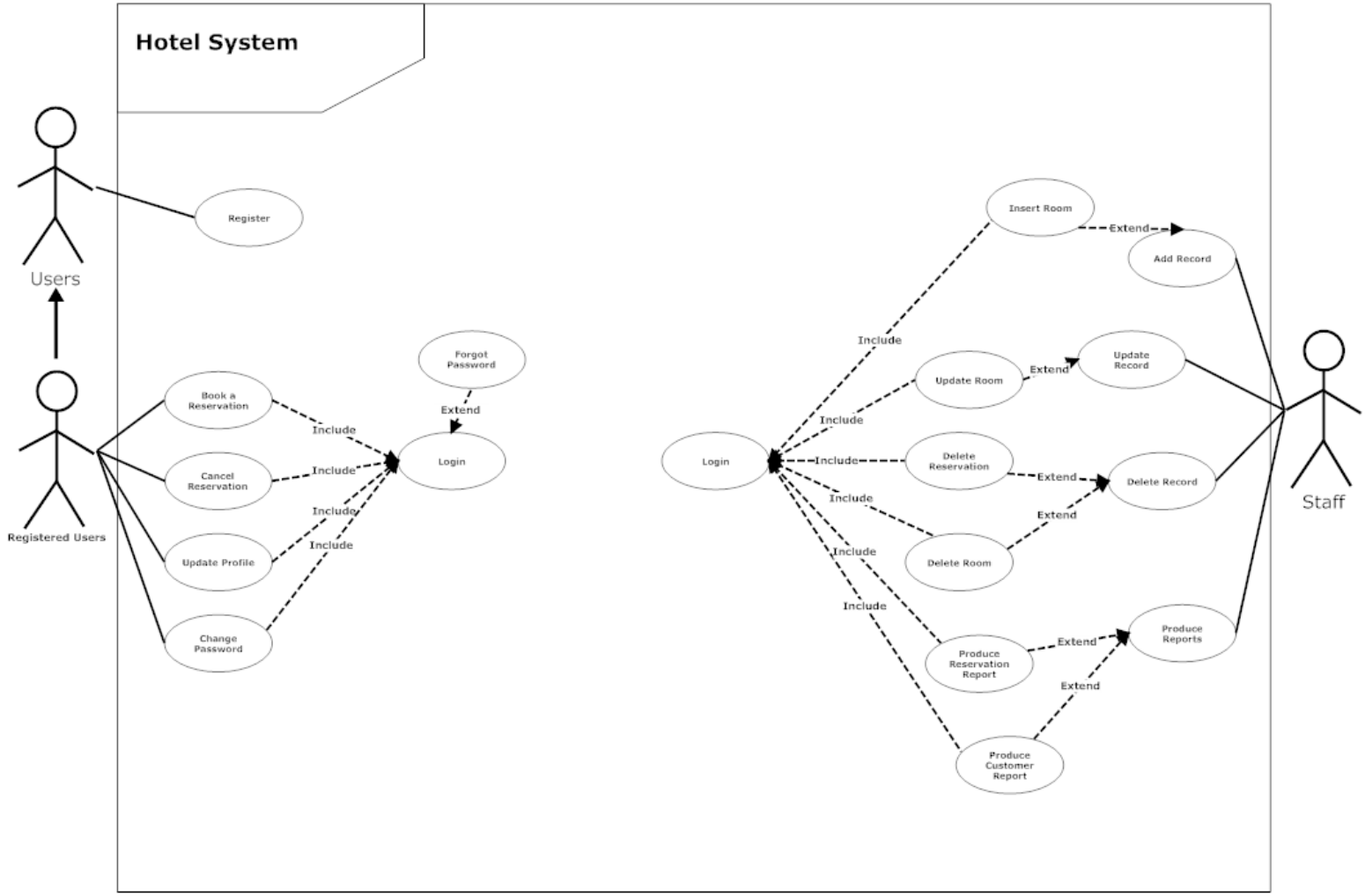


Figure 30 - System Use Case Specification

A use case diagram is essentially a diagrammatical design of the entire expected functionality of the system or parts of a system. The use case diagram is used to define the requirements of the system. A use case diagram must consist of four properties for it to be classified as such. These four properties include; Actors, Use cases, System boundaries and Associations. Actors are whoever or whatever expects a service from the system. They are usually a user role (staff, registered users etc.). On a use case diagram the actors will interact with the use cases. Actors are generally shown as a matchstick figure. The use cases will be shown as an oval shape and represent the different functions/activities or actions that the actor can carry out on the system. The system boundary is shown as a large box around all of the actors and use cases, to clearly indicate the boundary of the system. Finally the associations illustrate which actors will need to interact with which use case. This is shown by a line drawn, linking actors with the appropriate use cases. Often there are times when functions are repeated. When this occur the special association links; Include and Extend are used. They are shown by a dotted association line. Include is when the function must process the function it is pointing to before it is able to carry out its function. Whereas extend is where one function has the possibility of doing another function because it had carried out that function. Use case specifications on the other hand are detailed interaction between an actor and a use case. It is generally documented as use case description. This lists the steps that take place during the interaction. It is usually a textual description. To conclude use case diagrams are very helpful when trying to understand the functional business requirements. It is an excellent tool used to discuss the system and helps as a base to plan out the rest of the system.

Regarding the use case system diagram shown in figure 30, you can clearly see there are 2 main actors which will be interacting with this system; The User and the Staff member. The user in this case has only 1 function they are allowed to carry out on the system – this is to register. Once they have registered they become a Registered User and are able to carry out more functions. This is displayed by the arrow pointing from Registered User to User indicating that the Registered User inherits some properties from the User. This registered User is then able to carry out additional functions such as; Book a reservation, Update their profile and finally change their password. However, as you can see they are only allowed to carry out these functions once they have logged in. Additionally once a registered user has booked a reservation they have the option to cancel this reservation. The other actor, the staff , as shown in the use case diagram has many more functions in which this actor can carry out. They are able to delete records, update records and finally add records from/to reservations or rooms. Again, the staff must also be logged in to be able to carry out these functions.

The use case diagram clearly tells us that the system must have a way to indicate which actor is able to carry out which functions. Therefore, this suggests that the system needs to have a role function, to distinguish a normal user from an admin staff. However, like many other key components of a system this cannot be illustrated in this use case diagram. As a use case is only a representation of the requirements the system must be able to fulfil in accordance to the actors. Thus we must use other technique in being able to analyse and design the more complicated processes of the system.

3.4 Data Flow Model

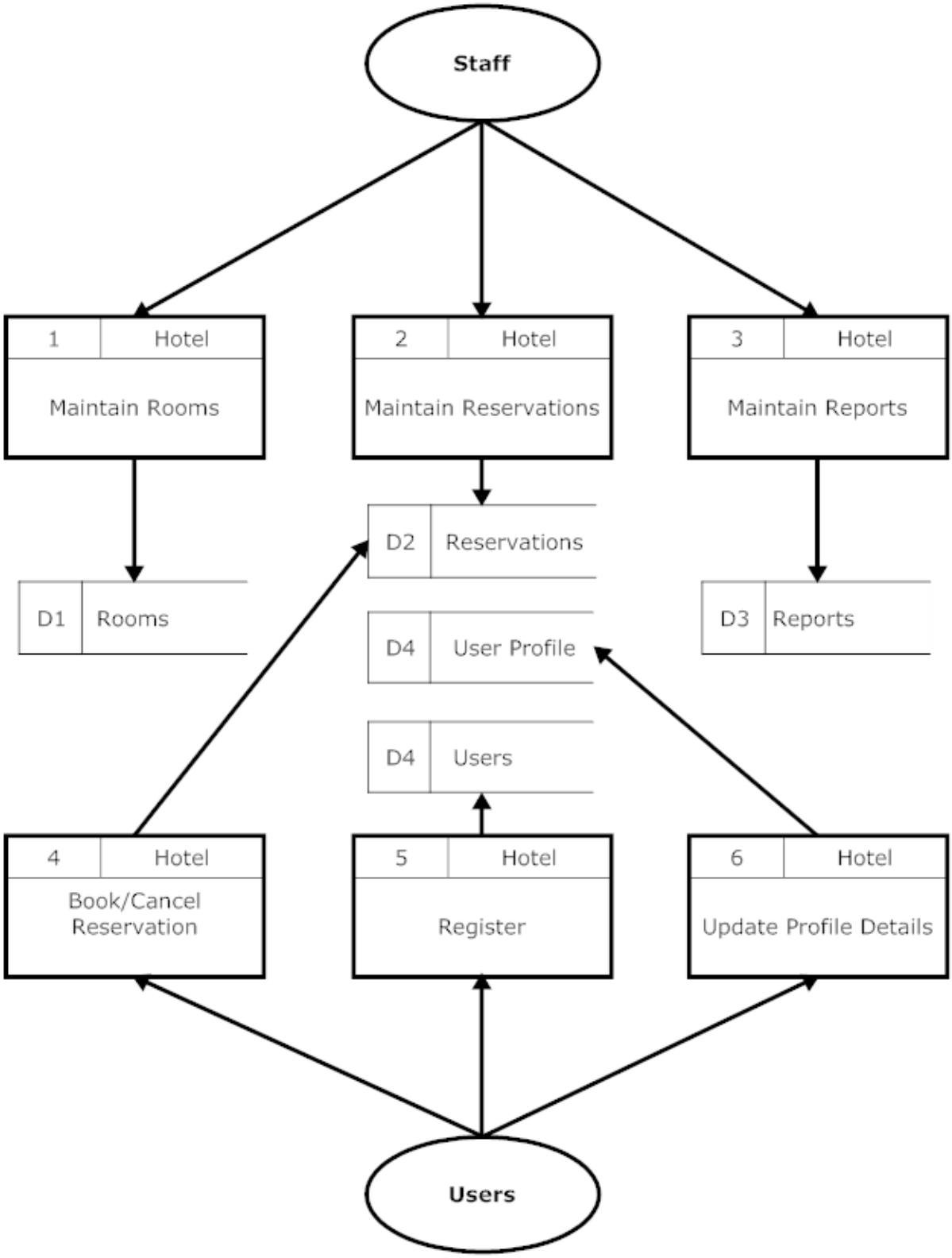


Figure 31 – Data Flow Diagram Hotel System

A data flow diagram is a useful technique to visually see information flows within a particular system. A data flow diagram shows us 3 important elements. How the information enters and leaves a system. What changes the information and finally where the information is stored. DFD's show the passage of data through a system by using 5 features: Data flows, Processes, Data Stores, External Entities and Physical Resources. Processes are transformations, changing the incoming data information into on-going data information. Processes are usually always drawn by a square box. The top of the box will have an identification number on the left and the role carrying out the work. In the middle of the processes is a descriptive name outlining what this process does. The data stores is where the information will be kept, this is usually seen as a table inside a database. Data stores are usually represented by an open ended box. Again, this box contains a meaningful name in the middle descripting the data store. It is also given an identification letter& number. Data flows are shown by arrows and they simply show the follow of data from the source to the destination, from/to the process to/from the data store. Finally the external entity, represents a person or part of an organisation, this person will be sending or receiving the data. Along with data stores they can be duplicated to simplify the presentation and understanding of the diagram.

For example you can see in Figure 31 that there is an external entity described as users. This user will send some information to the data process, as described this data process is updating some profile details which the user is sending. This data process then has a data flow to a data store outlined User Profile, which indicates the user has updated the profile details, through this process, the information then updates the data store which we know is a table in a database.

3.5 Data Dictionary

Table/Column Name	Data Type	Nullable	Primary/Foreign Key	Description	Example
Table_Users:					
User ID	UniqueIdentifier	NO	PK	ID of User	32 chars long
UserName	nvarchar(256)	NO	-	Users Name	kkhan
Table_Membership:					
User ID	Unique Identifier	NO	PK/FK	ID of User	32 chars long
Password	nvarchar(128)	No	-	Encrypted Password	28 chars long encrypted
PasswordFormat	Integer	No	-	Format	1
PasswordSalt	nvarchar(128)	No	-	Encrypted Password	28 chars long encrypted
Email	nvarchar(256)	YES	-	Email of Users	@gmail.com
Password Question	nvarchar(256)	YES	-	Secret Question	Favourite Colour
Password Answer	nvarchar(128)	YES	-	Secret Answer	Black
IsApproved	Bit	NO	-	Check	1
IsLockedOut	Bit	NO	-	Check	0
CreateDate	Datetime	NO	-	Date	2012-09-09 11:05:19
LastLongInDate	Datetime	NO	-	Date	2012-09-09 11:05:19
LastPasswordChangedDate	Datetime	NO	-	Date	2012-09-09 11:05:19
LastLockedoutDate	Datetime	NO	-	Date	2012-09-09 11:05:19
Table_Roles:					
Role ID	Unique Identifier	NO	PK	ID of Role	32 chars long
Role Name	nvarchar(256)	NO	-	Name of Role	Admin
Table_UsersInRoles:					
User ID	Unique Identifier	NO	PK/FK	ID of User	32 chars long
Role ID	Unique Identifier	NO	PK/FK	ID of Role	32 chars long
Table_UsersProfile:					
User ID	Unique Identifier	NO	PK/FK	ID of User	32 chars long
FirstName	nvarchar(20)	YES	-	Name of User	Kamran
LastName	nvarchar(20)	YES	-	Name of User	Khan
PhoneNo	nvarchar(20)	YES	-	Number of User	123
HouseNo	Integer	YES	-	No of House	12
StreetName	nvarchar(20)	YES	-	Street Name	Green
City	nvarchar(20)	YES	-	City Name	Manchester
Country	nvarchar(20)	YES	-	Country Name	United Kingdom
PostCode	nvarchar(20)	YES	-	Chars	WA42 0ds
Email	nvarchar(50)	YES	-	Email	@gmail.com
Table_Rooms:					
Room No	Integer	NO	PK	No of Room	101
RoomType	nvarchar(20)	YES	-	Type of Room	Single
PricePerNight	Money	YES	-	Price of Room	150
Maximum Occupancy	Integer	NO	-	Maximum	1

NoOfBeds	Integer	NO	-	No of Beds	1
NoOfBathrooms	Integer	NO	-	No of Bathrooms	1
Entertainment	Bit	NO	-	Check	1
RoomService	Bit	NO	-	Check	1
Gym	Bit	NO	-	Check	1
Table_Reservations:					
ReservationID	Integer	NO	PK	Id of Reservation	1
UserID	Unique Identifier	NO	FK	ID of Role	32 chars long
Room No	Integer	NO	FK	No of Room	101
CreateInDate	Date	NO	-	Date	2012-09-09
CreateOutDate	Date	NO	-	Date	2012-10-10
NoOfDays	Integer	NO	-	Number	30

Figure 32 – Data Dictionary Hotel System

The purpose of a data dictionary is to list all the tables which will be contained in a database and define the attributes which will be in each table in a database format. For each attribute in a table we must define some elements of these attributes so that it is easier for someone to develop the database. These elements include; the data type, define whether the attribute can be null or not, decide if the attribute is a primary key or foreign key of the table, a small description outlining the purpose of the attribute and finally an example of the attribute in question.

3.6 Relational Database Analysis

UNF	1NF	2NF	3NF
UserID	UserID	UserID	UserID
Password	Password	Password	Password
FirstName	FirstName	FirstName	
LastName	LastName	LastName	UserID
PhoneNo	PhoneNo	PhoneNo	FirstName
HouseNo	HouseNo	HouseNo	LastName
StreeName	StreeName	StreeName	PhoneNo
City	City	City	HouseNo
Country	Country	Country	StreeName
PostCode	PostCode	PostCode	City
Email	Email	Email	Country
RoomNo			PostCode
RoomType	UserID	UserID	Email
PricePerNight	ReservationID	ReservationID	
MaximumOccupancy	CheckInDate	RoomNo	RoomNo
NoOfBeds	CheckOutDate	CheckInDate	RoomType
NoOfBathrooms	NoOfDays	CheckOutDate	PricePerNight
Entertainment	RoomNo	NoOfDays	MaximumOccupancy
RoomService	RoomType		NoOfBeds
Gym	PricePerNight	RoomNo	NoOfBathrooms
ReservationID	MaximumOccupancy	RoomType	Entertainment
CheckInDate	NoOfBeds	PricePerNight	RoomService
CheckOutDate	NoOfBathrooms	MaximumOccupancy	Gym
NoOfDays	Entertainment	NoOfBeds	
	RoomService	NoOfBathrooms	UserID*
	Gym	Entertainment	ReservationID
		RoomService	RoomNo*
		Gym	CheckInDate
			CheckOutDate
			NoOfDays

Figure 33 – Relational Database Analysis (RDA) Hotel System

Normalisation or relational database analysis with regard to relational database design is the process of organising the data to minimize any redundancy. Normalisation or RDA usually involves dividing a database into multiple tables and defining the relationships between these tables. The purpose of this process is to allow the maintainability of a database in the long term. Therefore, the objective would be to separate the information so that any additions, deletions and modifications of an attribute/field can be made in just one field. There are 4 different forms of relational database analysis/normalisation;

Un-normalised Form (UNF): The purpose of UNF is to list all the fields/attributes to be stored in a database and assign a primary key to them.

As you can see from Figure 33 under UNF I have listed all of the possible attributes which will be needed for my database. I have also defined a primary key (User ID) to uniquely identify these attributes.

First Normal Form (1NF): First normal form is the process of removing any duplicates or repeating fields and attributes, then to assign these fields to another table within the database. Finally, to give the new table the same primary key as UNF and also assign a new primary/foreign key to the new table created.

In the same figure you will see under 1NF that I have split the existing single table from UNF into 2 tables. This is because the attributes in the second table I believe will be repeating, such as there will be multiple rooms and multiple reservations of the same room. Thus, I have given it the same primary key as well as assigning a new primary key (Reservation ID).

Second Normal Form (2NF): The rule for the second normal form is to find any tables with a partial dependency to the primary key. Once this is found you would then split these tables and assign this partial dependency key to the primary key of a new table and keep this key in the old table as a foreign key.

As you can see from the figure33, it concludes that the reservation id is partially dependant on room id. Therefore, the table has been split into a Reservation table holding the UserID, ReservationID and RoomID, and then a new room table has been created which is given RoomID as a primary key.

Third Normal Form (3NF): Third normal form is to find any non-key fields/attributes that depend on another non-key attribute and if there is to remove this. As this, is redundant data which can be retrieved via the primary key.

Finally there was no non-key field which was not independent. So these tables have been moved over to 3rd normal form as no extra alteration was needed.

3.7 Entity Relationship Model

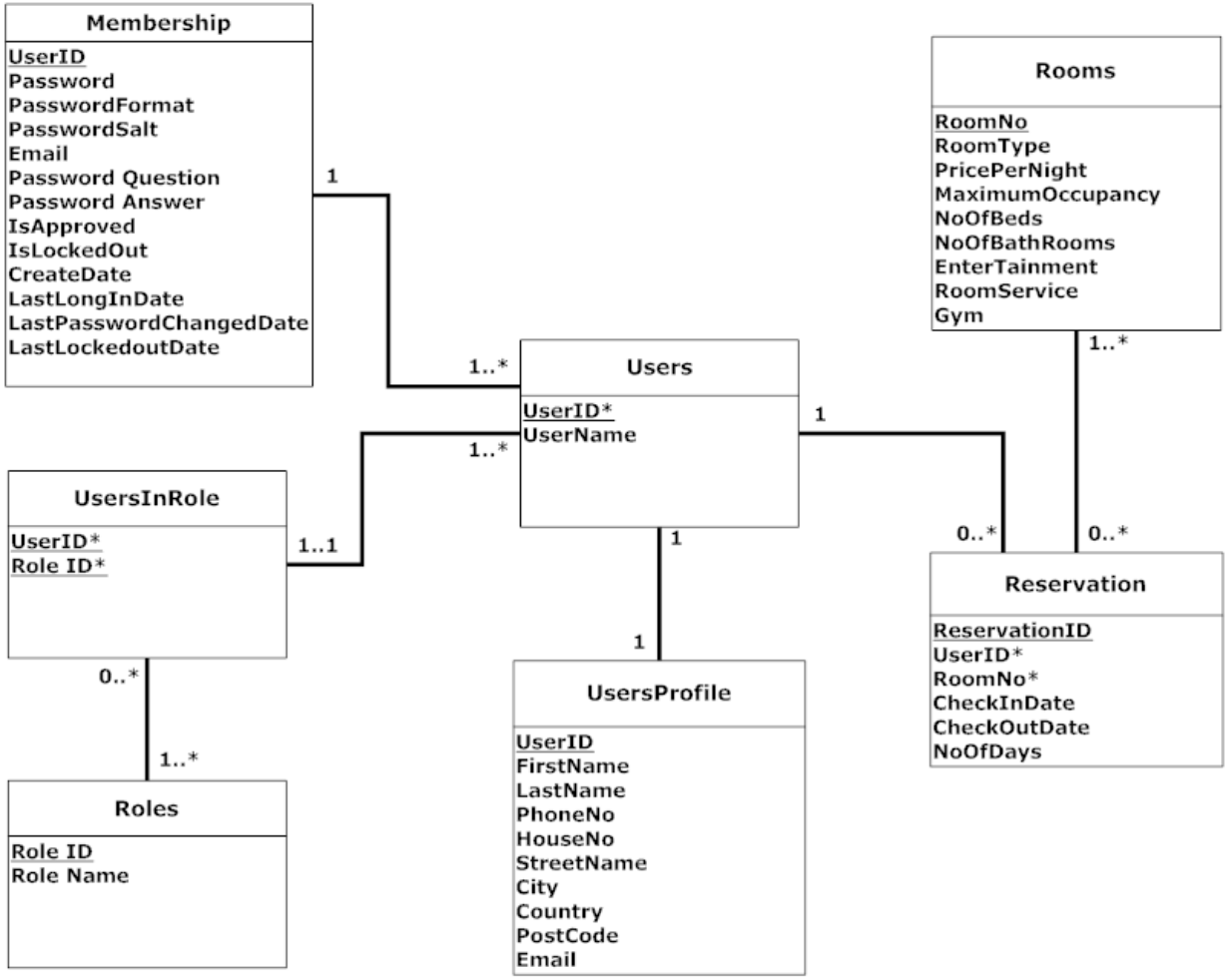


Figure 34 – Entity Relationship Diagram Hotel System

An entity relationship diagram is a diagrammatical model of the data which will be stored in a system. This type of data model allows the stakeholders who will be using the system to agree on the data that will be recorded and retrieved from the system. ER diagrams can also be used as a building block to produce databases. However, it has many other uses such giving a better understanding of the data required and provides mechanism to moving the data required, to a design and eventually the development of the system. Entity relationship diagrams or any data models are primarily concerned with outlining and understanding 3 elements; data items, grouping of attributes (entities) and finally the relationship between these entities. The entities are illustrated on the ER diagram by a box. Each entity will have an appropriate name. The entity will be the base of a table in a database, each entity must contain attributes with at least one unique identifier (primary key). As mentioned the entities will contain attributes (data items). These attributes do not have to be unique to the entity itself, we call these types attributes, foreign keys. The relationship is the connection between two entities, these relationships are represented on this diagram by a line going from one entity to the other. The relationships may come in the form of one-to-many, one-to-one or many-to-many.

In figure 34 you can see the entity relationship diagram for the proposed hotel system. In this diagram there are 7 different entities of the system. The entities and the attributes in them have been derived from the collected requirements which have been gathered through the use of a use case diagram to produce this system. As you can see each entity has a unique identifier or primary key which is associated with the entity. Some entities such as the user profile will have a foreign key and a primary key assigned to the same attribute. If you take a look at the numbers between Users and the UserProfile entity, it will indicate to you the relationship between these entities. In this particular case, the User is associated with 1 instance of a UserProfile and a UserProfile is associated with 1 or many User entities.

3.8 Class Model

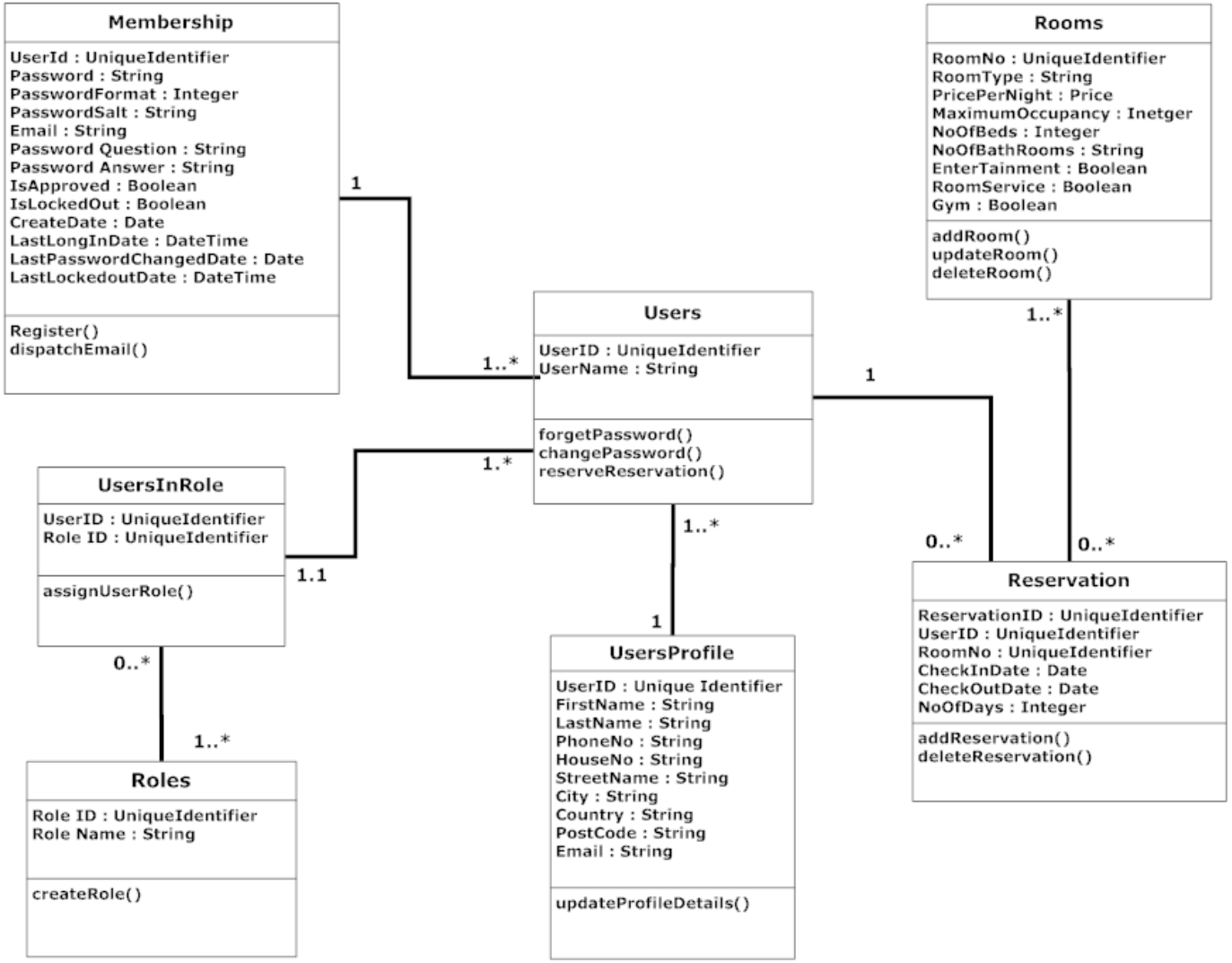


Figure 35 – Class Model Hotel System

A class diagram is the unified modelling language (UML) way of showing graphically the data and functions that will be a part of the system. This model has many similarities to the entity relationship diagram and in most cases applies the same techniques. Again, the class diagram has to have some common elements for it to be considered a class diagram. A class diagram must show the relationship between the different classes/objects in the system, their attributes and the operation that will be carried out. On occasions multiplicities and generalisations/specifications would be applicable for a system but it is not needed to have a class diagram of a system. The operations are stored at the bottom level of a class diagram as shown in figure 35. These operations are invoked via communication being sent to the class by other classes. These operations are always known as methods. The attributes are held within the class and are only available to that particular class unless generalisation is being used. This is a form of object orientated principle, encapsulation, where data is restricted from other classes.

From the class diagram one can identify that the user must register by entering in an email and password. Once the user has registered an email, it will be dispatched to this user who will then confirm their registration. Once this has been done a permanent user class will be granted to this person. They will then have access to the User Profile object class which they will be able to update their profile details, as well as change their password. A role will be assigned to this user automatically usually of low security level. The user will also be able to make reservations and cancel any reservations associated with them, now that they are a registered member.

Similarly if an admin logins in, they will have more control. They are able to create roles. Assign people to these roles. They are also able to manipulate rooms by adding a new room, deleting an existing room or updating information of an existing room.

3.9 Wireframe Designs

Storyboarding or wireframe designs as they are formally known in system design and development. Is a good way, to initially design the look and functionality of the system. As it, provides a visual representation of the final prototype of the system. Wire framing can show many features of the expected system including the low level functionality, therefore, it’s not just limited to the aesthetical look of the system.

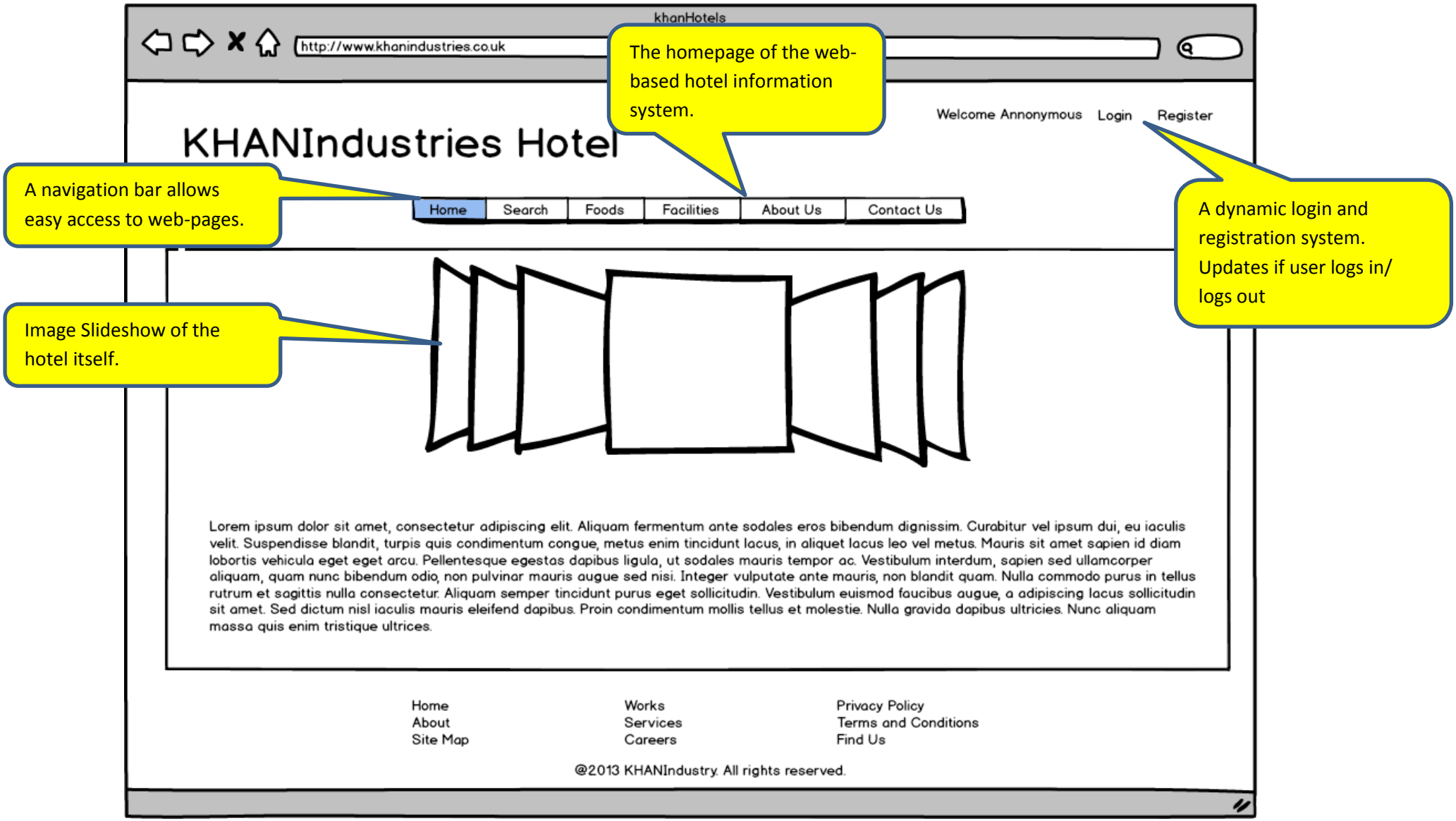


Figure 36 – Wireframe Home Page

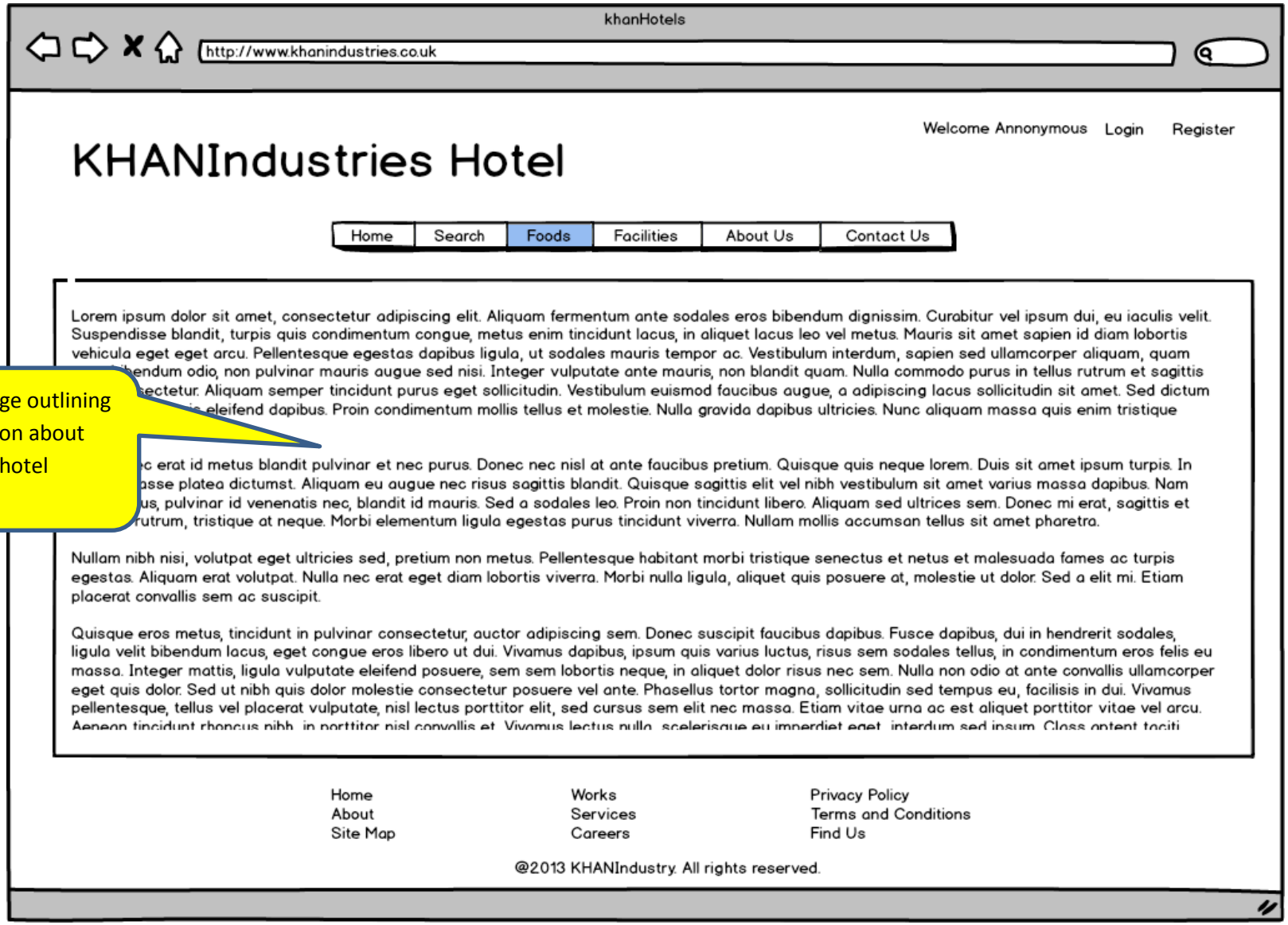


Figure 37 – Wireframe Foods Page

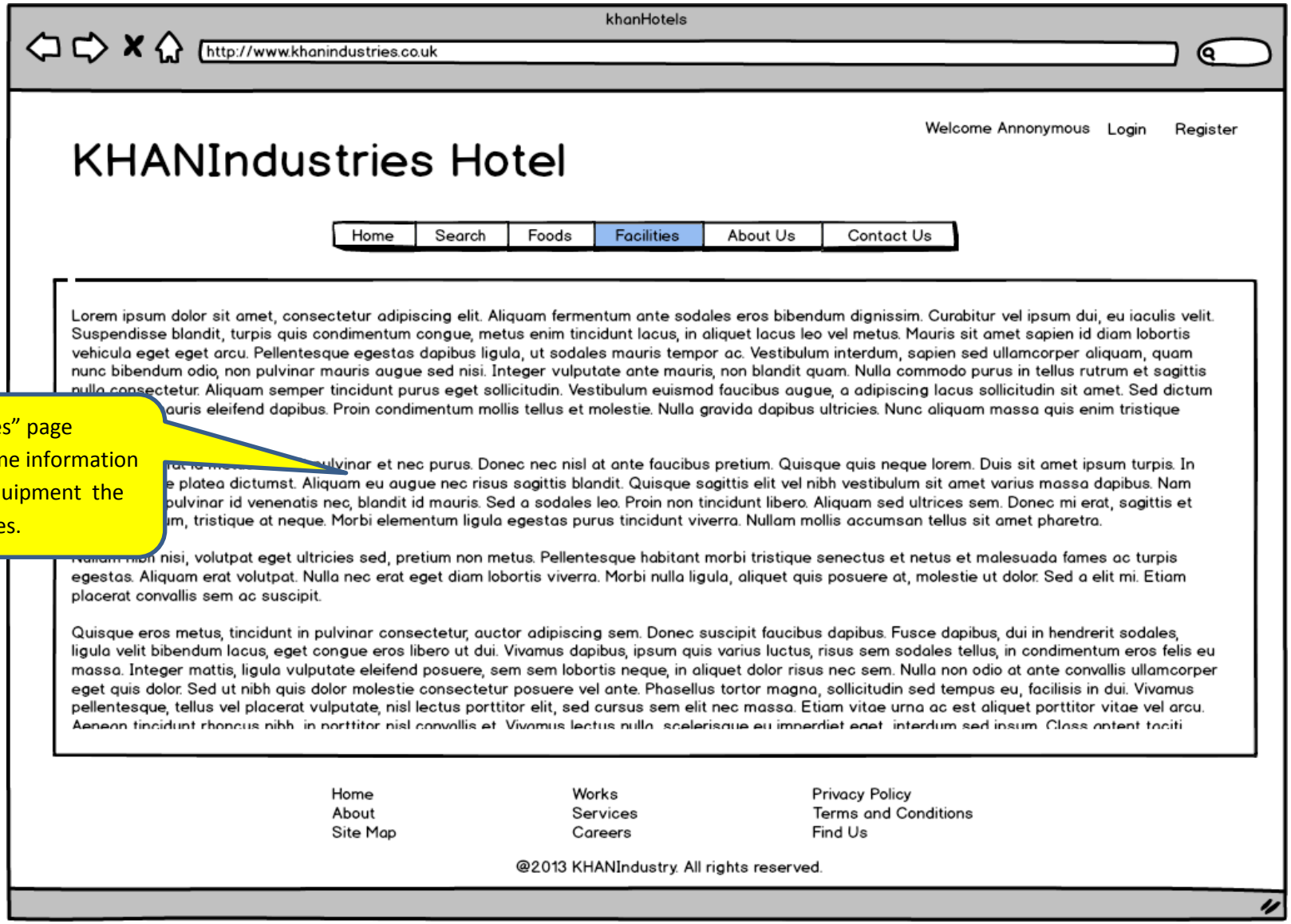


Figure 38 – Wireframe Facilities Page

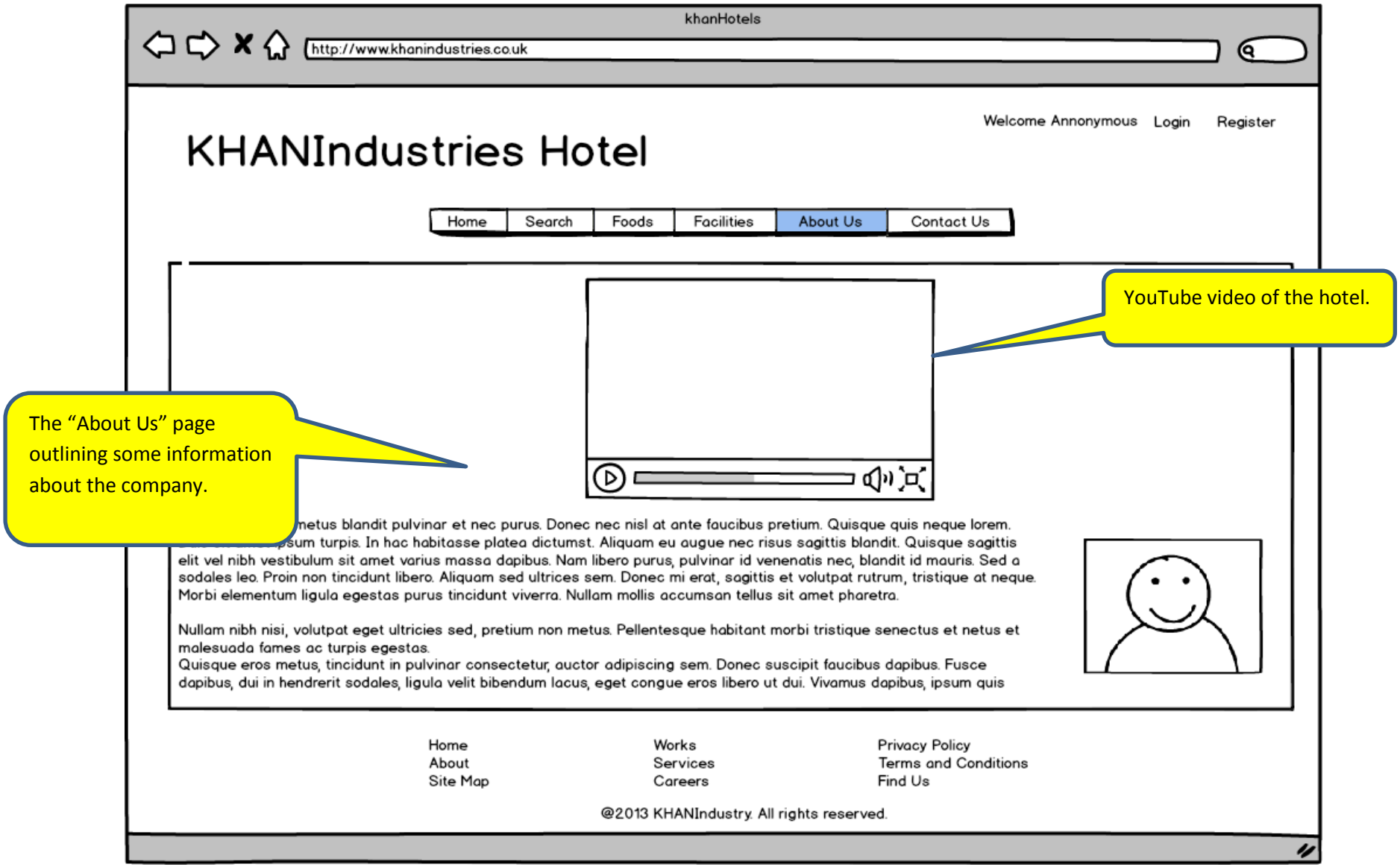


Figure 39 – Wireframe About Us Page

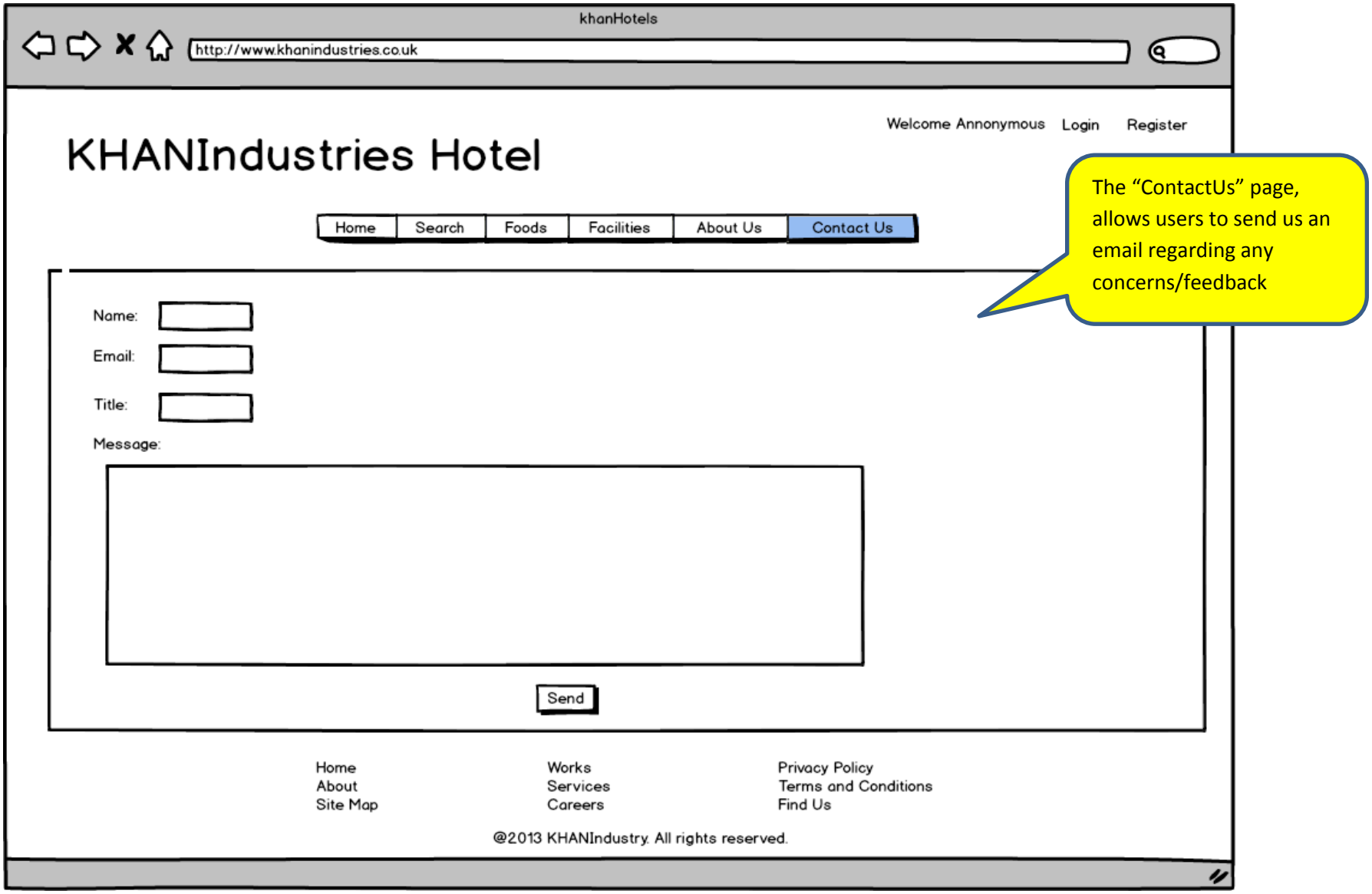


Figure 40 – Wireframe Contact Us Page

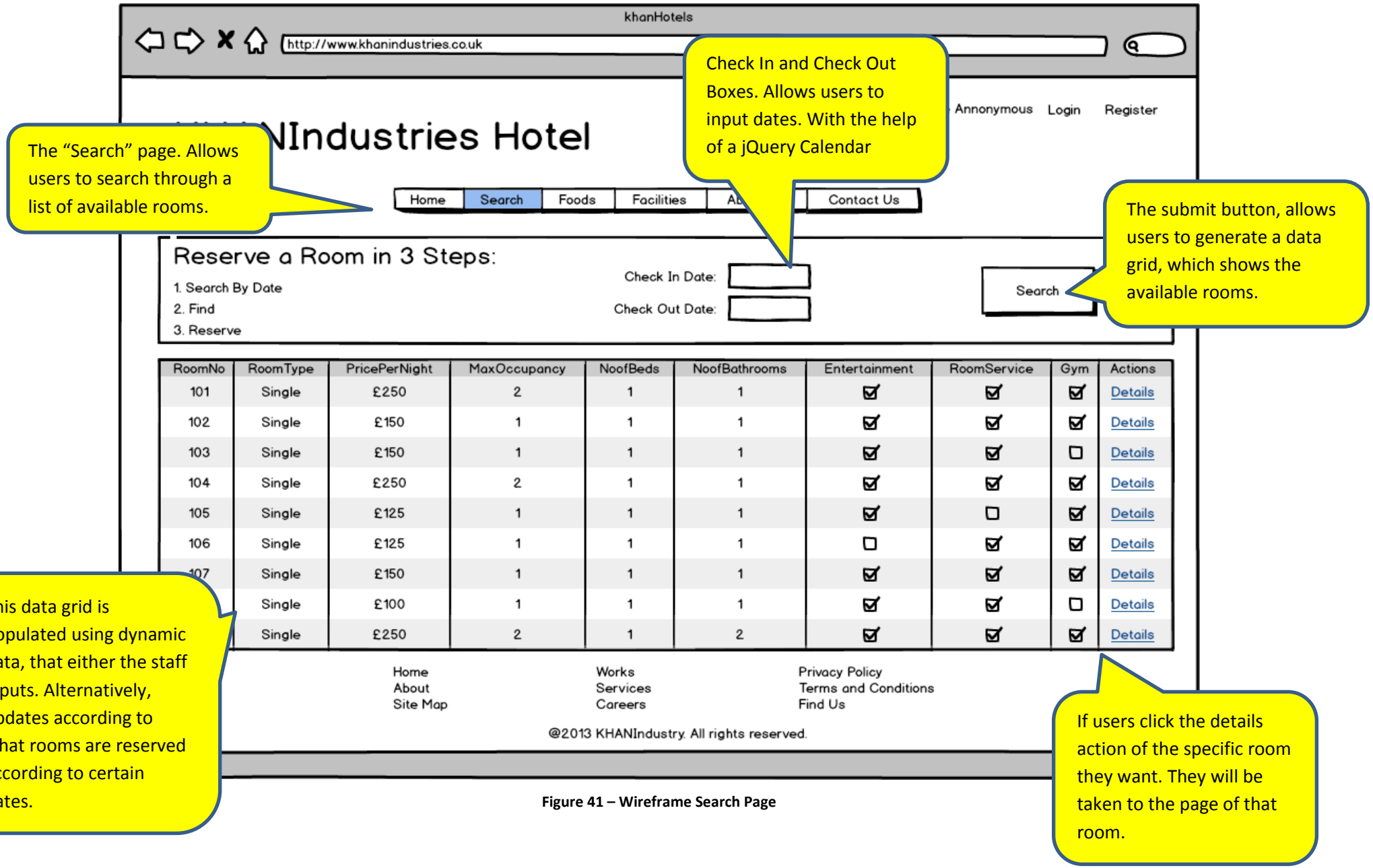


Figure 41 – Wireframe Search Page

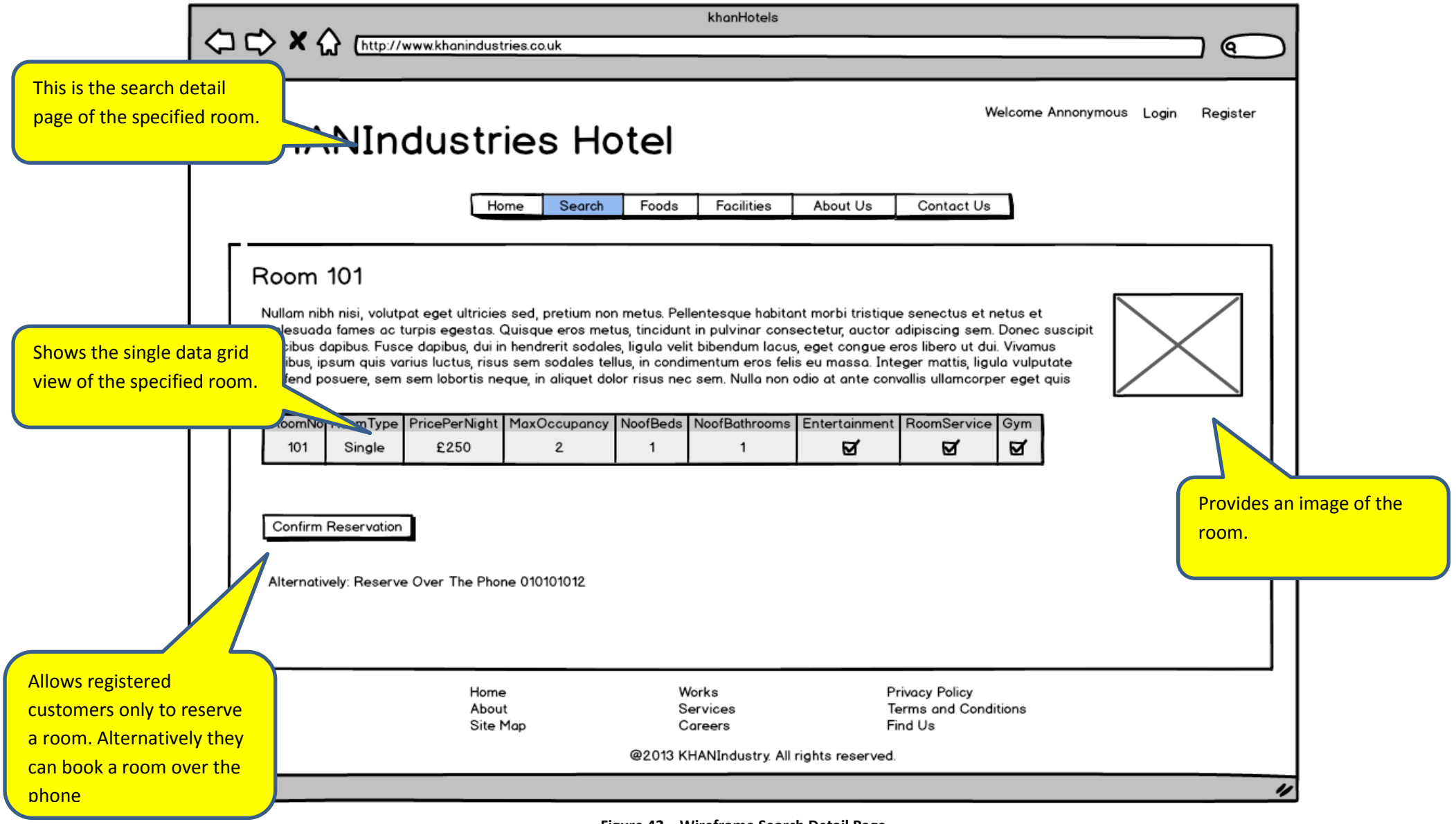


Figure 42 – Wireframe Search Detail Page

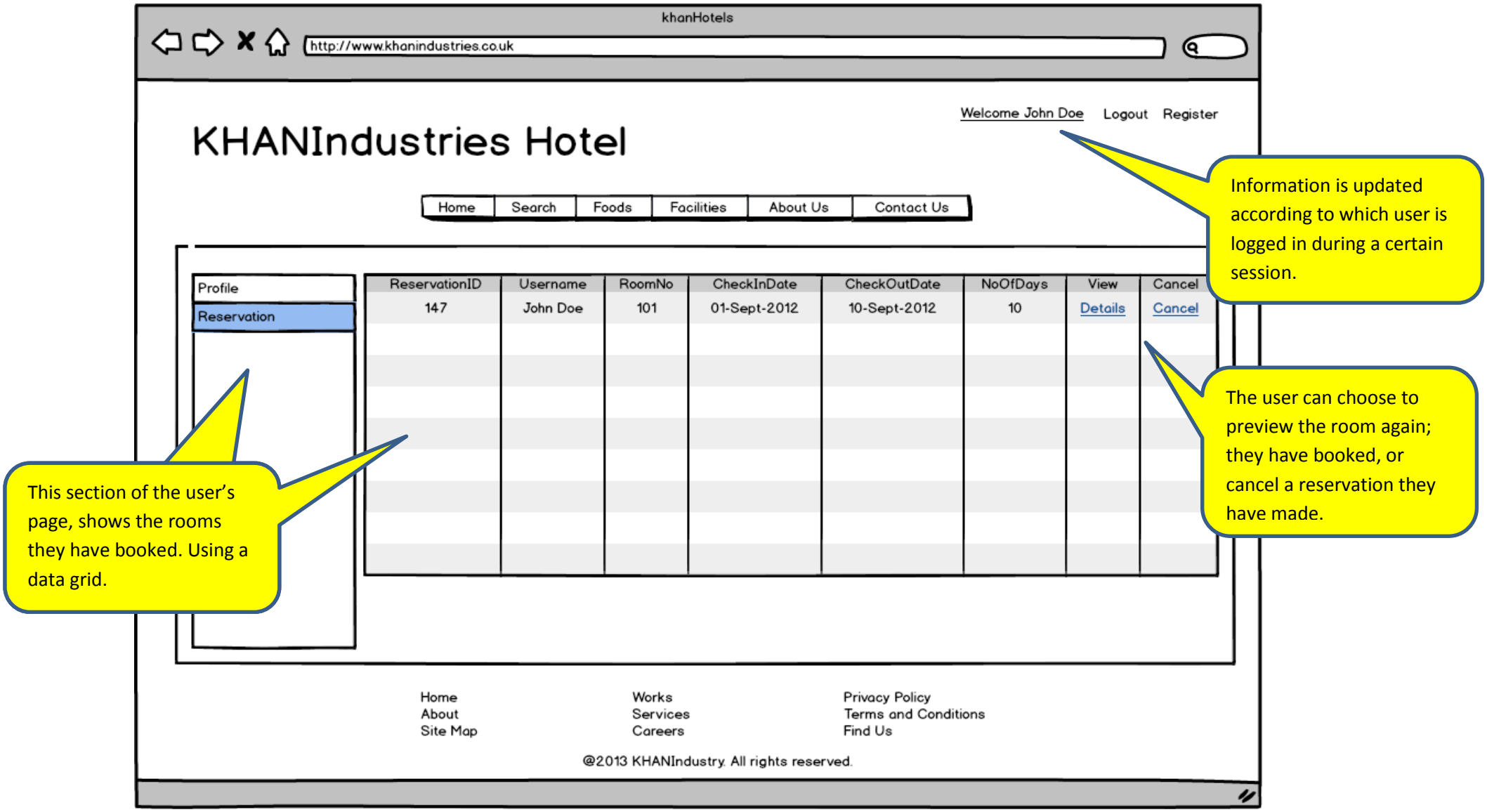


Figure 43 – Wireframe User Profile Reservation Page

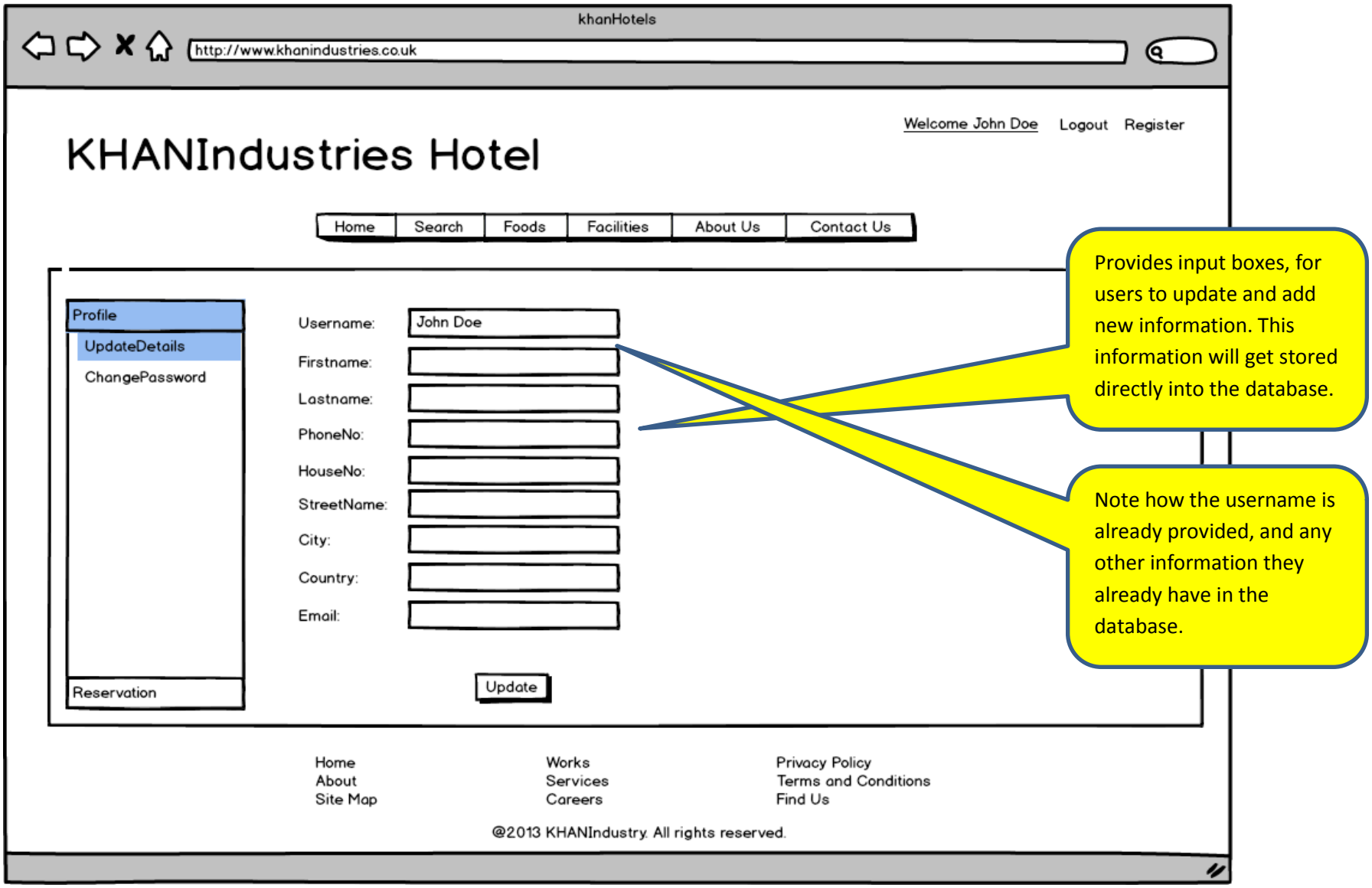


Figure 44 – Wireframe User Profile Update Details Page

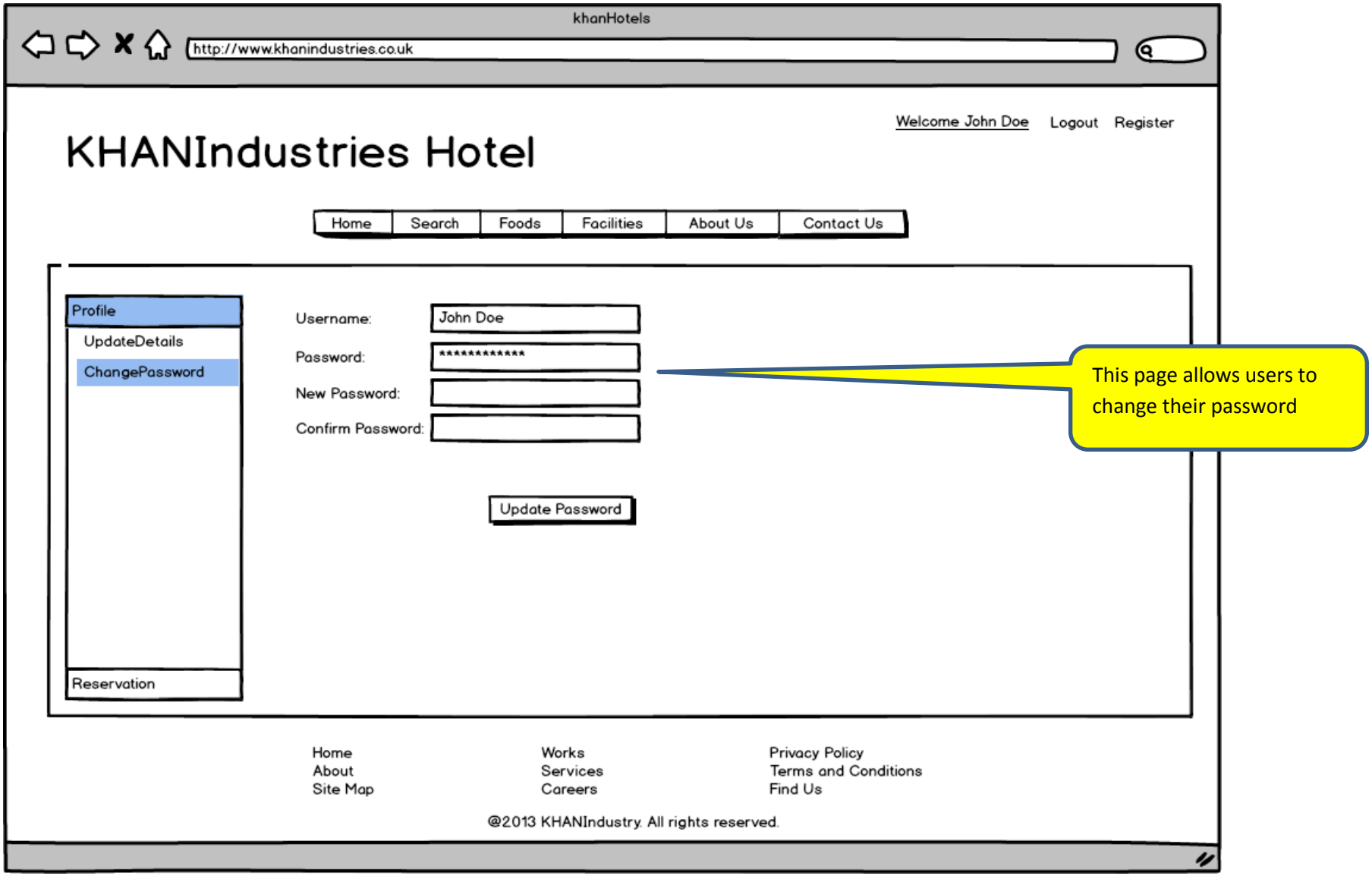


Figure 45 – Wireframe User Profile Change Password Page

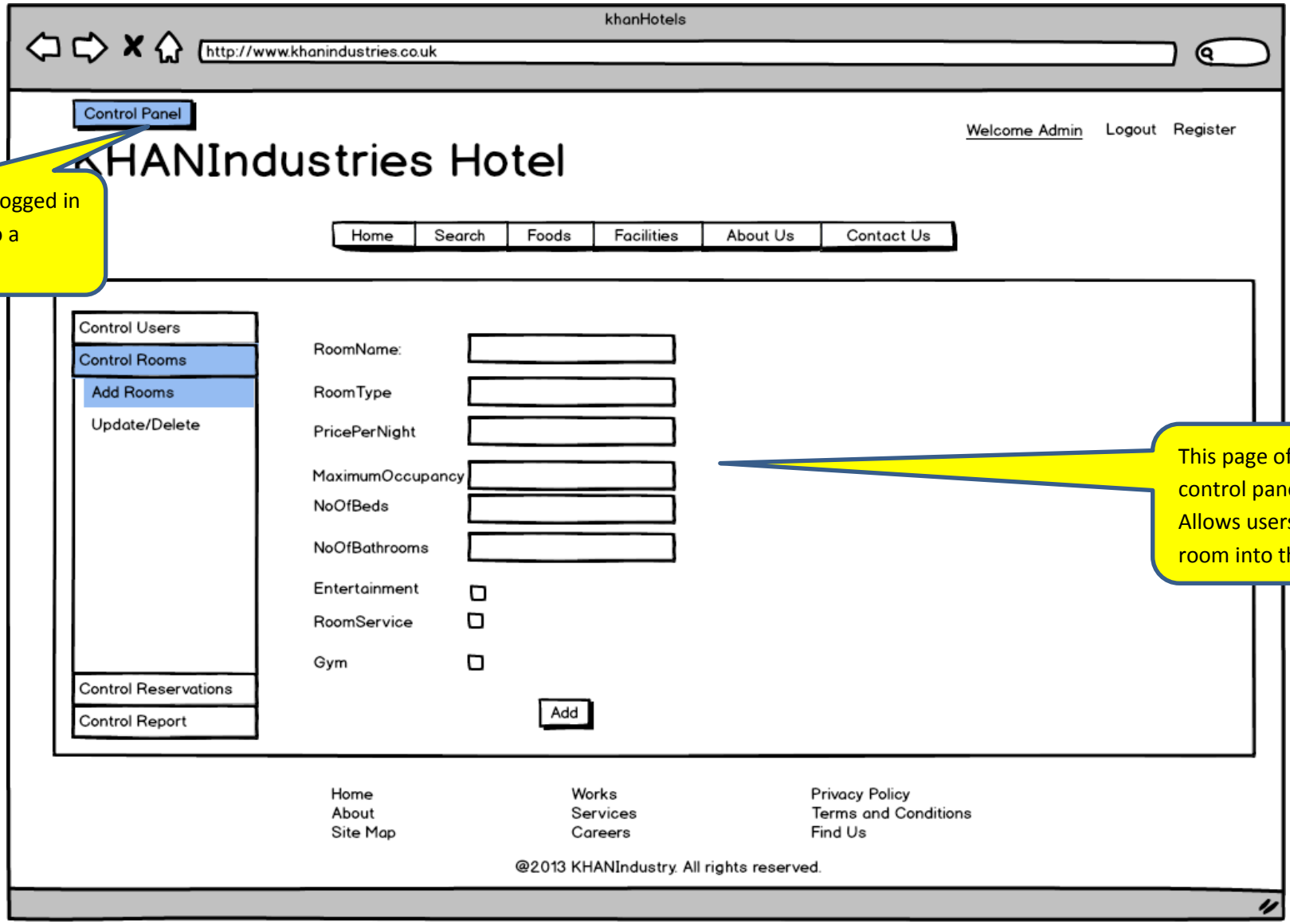


Figure 46 – Wireframe Admin Add Rooms Page

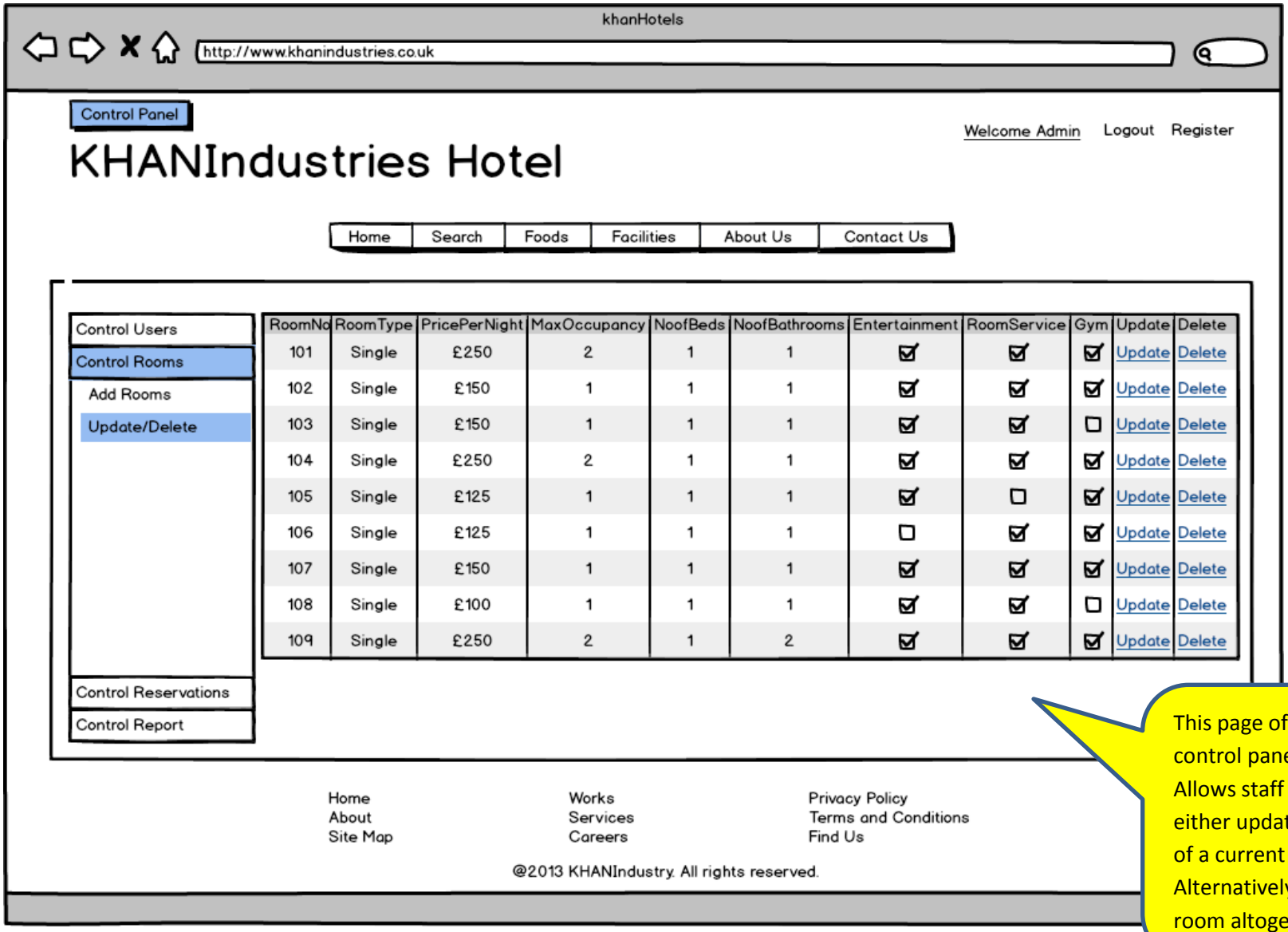


Figure 47 – Wireframe Admin Update/Delete Rooms Page

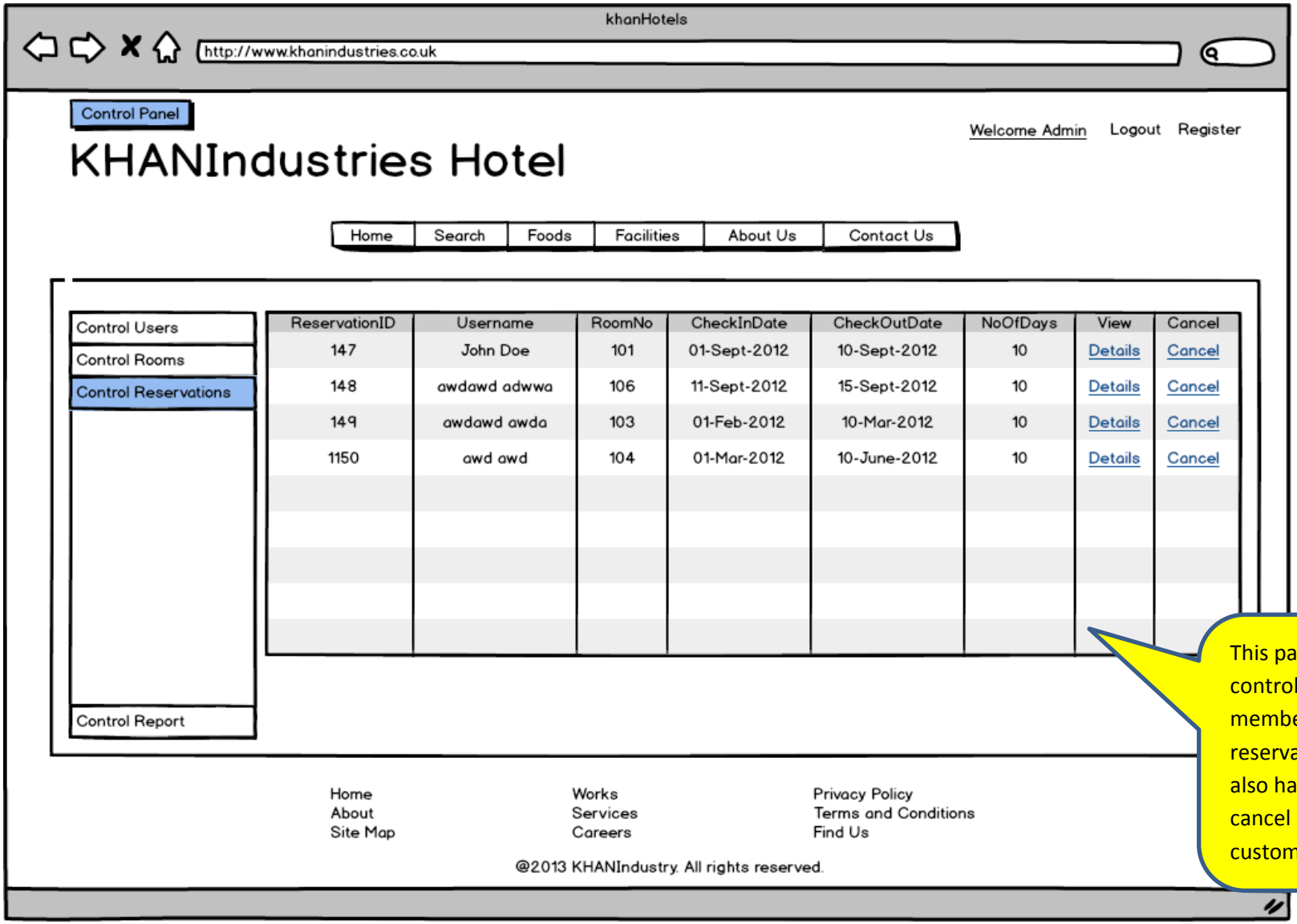


Figure 48 – Wireframe Admin Control Reservation Page

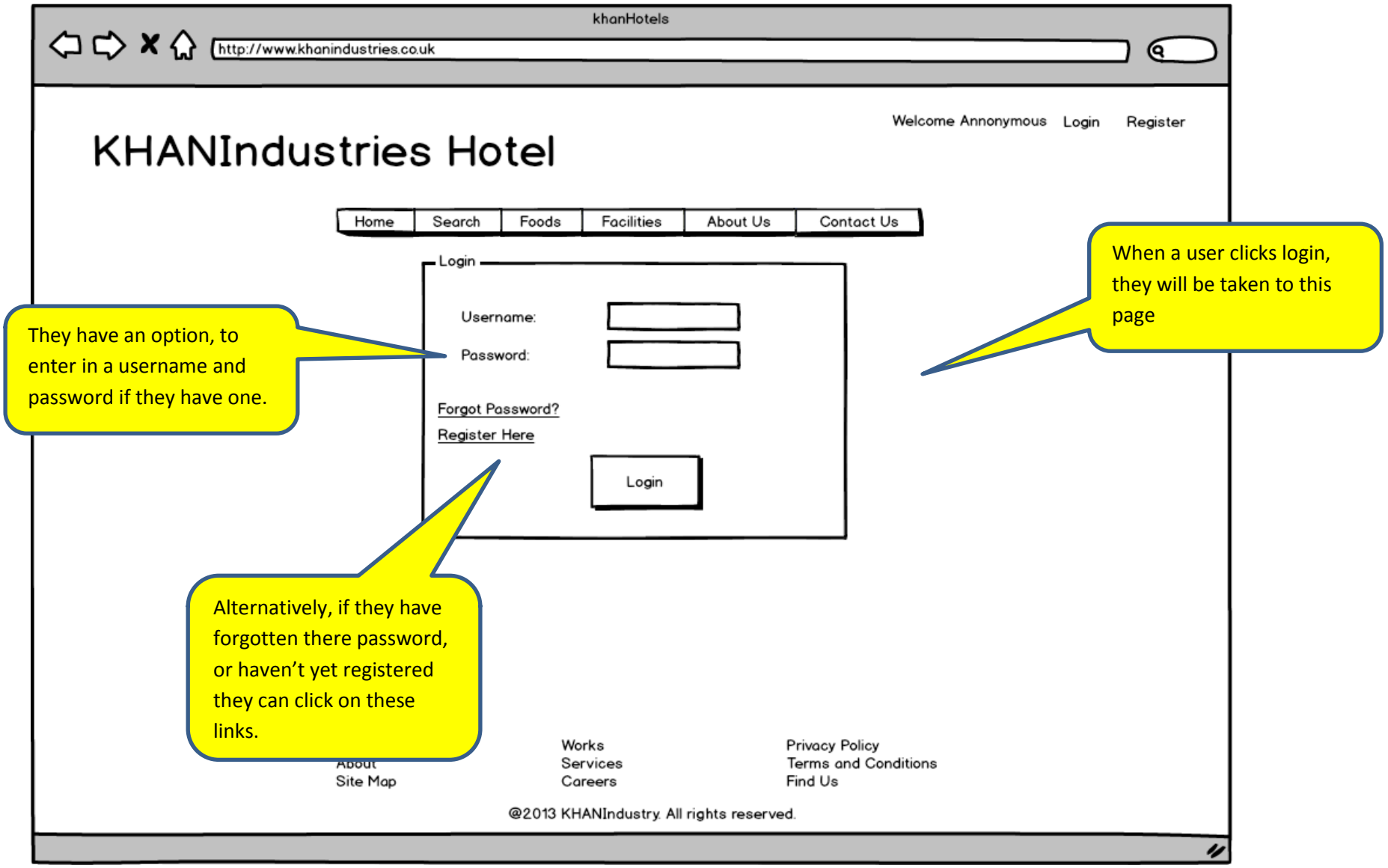


Figure 49 – Wireframe Login Page

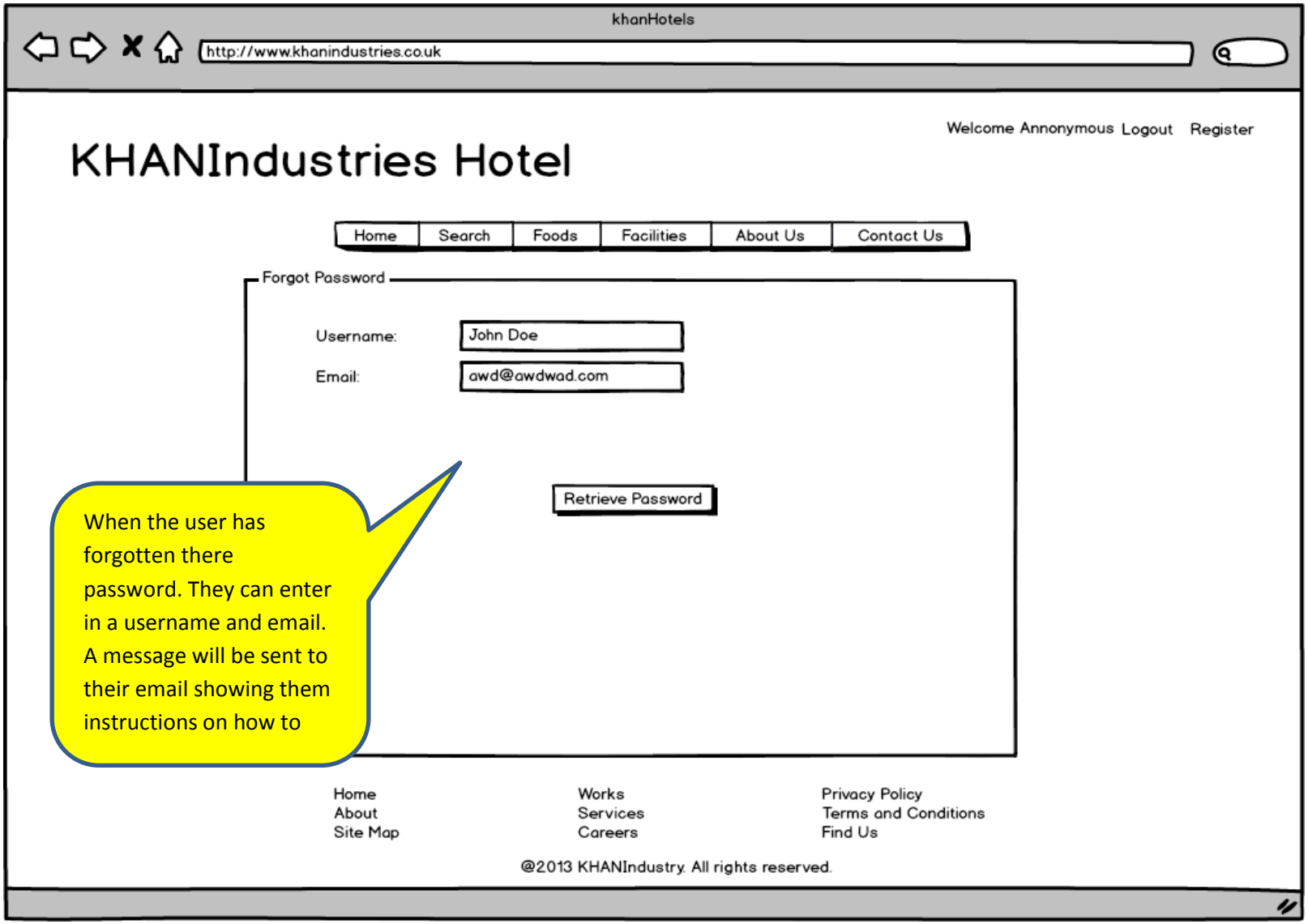


Figure 50 – Wireframe Forgot Password Page

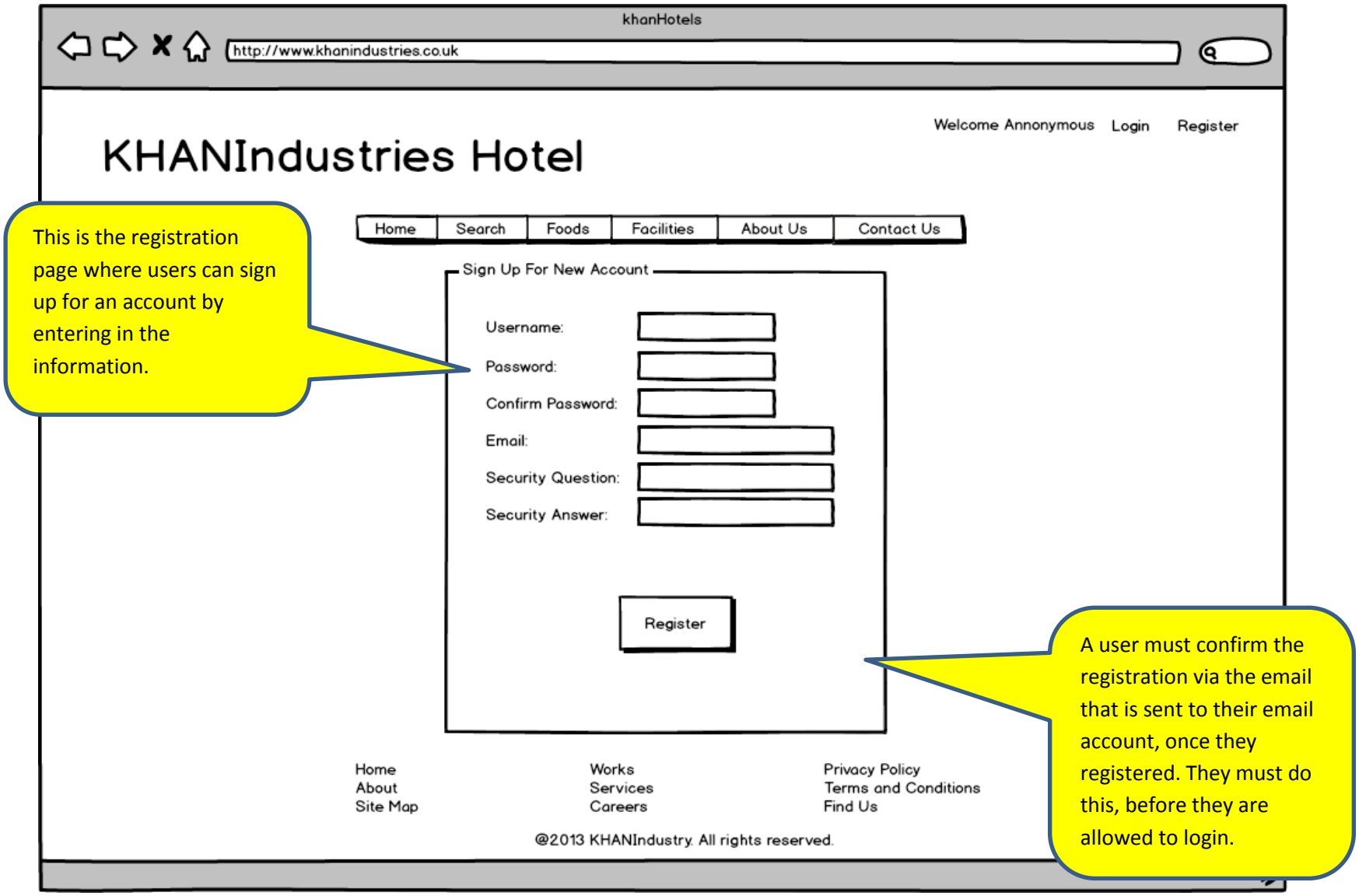


Figure 51 – Wireframe Registration Page

3.10 Summary

To summarise, the above chapter demonstrated the common techniques used to analyse and design a system. These techniques were compiled from the literature survey. The chapter discussed and described the functionality of the expected system, the model of the expected system in terms of the business functions and also the technical functions and finally the prototype designs of the intended system were produced.

The purpose of the chapter was to carry out the analysis and design techniques to allow one to build a more efficient information system. Now that the structure of the system has been outlined, the development phase/implementation phase can now be conducted. As there is enough information to go off, to develop every aspect of the system. Additionally, with the completion of this chapter we have also completed the analysis and design objectives outlined in chapter 1.

4 Systems Development

4.1 Introduction

The systems development chapter describes the implementation of the proposed system. This chapter will outline the steps taken to develop the systems functionality acquired and documented in chapter 3. A description of primarily the back-end development will be given. The back-end development for this system consists of all the functional requirements presented in chapter 3. The functions include user functions as well as admin function. Additionally database development will be outlined.

4.2 Database

As discussed and concluded in previous chapters of this dissertation, the database server being used to produce this system is Microsoft's SQL Server. This database server is accompanied with Microsoft's relational database management system (RDBMS) SQL Management Studio (SMS). SQL management studio provides assistance in managing the database regarding all aspects of development.

4.2.1 Setting up the database

Step 1 to developing such database is to describe/create the database itself using SQL management Studio. Initialising a database will enable one to add tables to the relational database, eventually inserting columns and finally adding values to the rows of the database.

```
USE [master]
GO
CREATE DATABASE [HotelProject]
CONTAINMENT = NONE
ON PRIMARY
(NAME = N'HotelProject', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.SQLEXPRESS\MSSQL\DATA\HotelProject.mdf' ,
SIZE = 5120KB ,
MAXSIZE = UNLIMITED,
FILEGROWTH = 1024KB )
LOG ON
( NAME = N'HotelProject_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.SQLEXPRESS\MSSQL\DATA\HotelProject_log.ldf' ,
SIZE = 8384KB ,
MAXSIZE = 2048GB ,
FILEGROWTH = 10%)
GO
```

Algorithm 1 – Creating The Database

Algorithm 1, as shown above presents the SQL Code for creating the database. Microsoft's SQL Management Studio produces two files which make up a single database. The first file being the .MDF file. The .MDF file costs of all the raw daw. The second file is the .LOG File. The .LOG file consts of all changes made to the .MDF file.

As the system will utilise ASP.NET, the system takes advantage of the ASPNETDB provided. To do this, the website administration tool had to be setup using Microsoft's Visual Studio. It is then made easy for someone to import the ASPNETDB.MDF file into the existing SQL Database being produced

as shown above. The purpose of using ASPNETDB is that it allows someone to manage securely every aspect of users using the system, such as, creating a new user, assigning their role and accessibility.

4.2.2 Creating Tables

Now that the database for the system has been created and the ASPNETDB Scheme is located in the same database. The Process of creating the tables required for this system will begin. The tables being produced are derived from the systems analysis and design in chapter 3 of this dissertation, specially the data dictionary and the entity relational model. When creating the tables, SQL management studio is utilised to achieve this. By producing the SQL queries for the create script shown below in Algorithm 2.

```
CREATE TABLE "Rooms"(
        RoomNo int NOT NULL,
        RoomType nvarchar(20) NULL,
        PricePerNight money NULL,
        MaximumOccupancy int NULL,
        NoOfBeds int NULL,
        NoOfBathrooms int NULL,
        Entertainment bit NULL,
        RoomService bit NULL,
        Gym bit NULL

        CONSTRAINT PK_Rooms PRIMARY KEY(RoomNo)
)
```

Algorithm 2 – Creating Example Table Rooms

The Algorithm presented above describes the structure of the create script for a specific table in our database, in this case it describes the “Room” table. All of the create tables are stored in a single script. Creating the tables outlines the columns for each table, which describes their data types for each column. Furthermore, any constraints in the tables are identified. For example, any Referential Integrity or Entity Integrity constraints. For example, as shown in Algorithm 2 above, it shows the entity integrity constraint of the column RoomNo in the table Rooms. This describes that the RoomNo is the primary/unique key of this table.

4.2.3 Inserting Values

Inserting values can be done just as easily using SQL management studio. As showing in Algorithm 3 Below.

```
INSERT Rooms(RoomNo,RoomType,PricePerNight,MaximumOccupancy,NoOfBeds,NoOfBathrooms,Entertainment,RoomService,Gym)
        VALUES(101,'Single',150,1,1,1,'true','true','true');
```

Algorithm 3 - Inserting Example Value Into Table

To be able to access the data from the database in Microsoft Visual Studio Website, we first need to establish a connection between the two. This is done by configuring the web.config file which is created automatically upon creating a website in visual studio.

```
<connectionStrings>
  <add name="HotelProjectConnectionString" connectionString="Data Source=KHAN-PC;Initial
Catalog=HotelProject;Integrated Security=True" providerName="System.Data.SqlClient" />
  </connectionStrings>
```

Algorithm 4 – Web.Config Establishing Connection To Database

Algorithm 4 describes the connection, which links the SQL Server and the Visual Studio Website. It does this by outline a connection string first and assigning its properties. Firstly it establishes the connection name as “HotelProjectConnectionString”, then goes on to establishing the connect string which locates the database by finding the data source and catalogue. Finally, it establishes the provider of the database, which by default in Visual Studio is SQL Server. This concludes the initial development and integration of the database.

4.2.4 User Functionality

The functionality which will be outlined below are taken from the requirements documented in chapter 3 of this dissertation. The functionalities include: Registration, Login, Forgot Password, Change Password, Update Profile, Search and finally book a room for reservation.

4.2.4.1 Registration

Registrations functionality pre-condition should be that inputted data by the user must be validated. The post-condition is that upon registration and authentication email must be sent to the user's email which they provided.

The system utilises a wizard to effectively carry out the registration function and generate the steps needed to successfully register.

```
<asp:CreateUserWizard ID="CreateUserWizard1" runat="server" DisableCreatedUser="True"
  OnSendingMail="CreateUserWizard1_SendingMail" OnCreatedUser="CreateUserWizard1_CreatedUser">
        <WizardSteps>
            <asp:CreateUserWizardStep ID="CreateUserWizardStep1" runat="server">
            </asp:CreateUserWizardStep>
            <asp:CompleteWizardStep ID="CompleteWizardStep1" runat="server">
            </asp:CompleteWizardStep>
        </WizardSteps>
</asp:CreateUserWizard>
```

Algorithm 5 – Registration Creation Wizard

Algorithm 5 represents a snippet of programming code taken from the much larger code to produce the registration functionality. With the use of the wizard the system can have multiple steps to producing this function. For registering a user only two steps are needed. As shown in Algorithm 5 the steps are CreateUserWizardStep and CompleteWizardStep. The first step simply lists the textboxes where the user is able to input their information in order to register an account with this system. As shown in Figure 52 below.

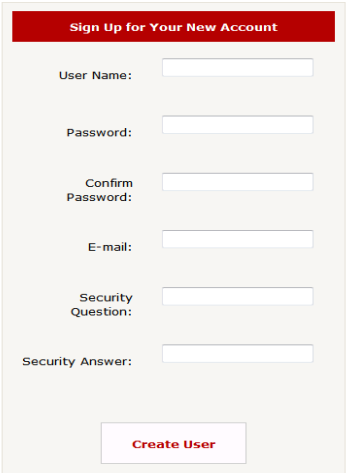


Figure 52 – Registration Function

The complete wizard step is the final step and just confirms the registration and requests the user to authenticate there details by clicking on a link sent to their email.

As mentioned previously, the pre-condition for the registration function states that the user must pass validation checks and ensure that the data inputted is correct.

```
<asp:TextBox ID="UserName" runat="server" Width="150px"></asp:TextBox>

<asp:RequiredFieldValidator ID="UserNameRequired" runat="server" ControlToValidate="UserName" ErrorMessage="User
Name is required." ToolTip="User Name is required." ValidationGroup="CreateUserWizard1">
*</asp:RequiredFieldValidator>
```

Algorithm 6 – Registration Function Pre-Condition Username

Algorithm 6 details specifically with the validation of the username, however, in the entire piece of code for the CreateUserWizardStep shows validation for ever field of the registration function. The first part to this algorithm, declares the textbox and assigns the ID to Username to distinguish it self from other textboxes. The second part of the algorithm defines any validation that the textbox requires. Required Field Validator control is used to achieve validation. The validator will first define the control it needs to validate in this case the control is the textbox, which has the ID of UserName. The Field Validator allows someone to provide an error message to show to the user if they failed validation. A ToolTip is also provided to help the user to correct the error and finally a “*” is appended to the textbox to indicate that there is an error here. This is a common practise when validating data.

The post condition for the registration function states that the user must first authenticate their details via the email they entered before the user is allowed to login. Refer to Algorithm 5, the CreateUserWizardStep allows one to define extra properties for certain events, two events are defined as OnSendingMail and OnCreatedUser. The events defined here point to the functions being programmed in C# in the .aspx.cs file which is a part of the Registration.aspx file. Algorithm 7, defines the mail definition. A mail definition allows us to send an email to the user by defining some properties of the email being sent.

```
<MailDefinition BodyFileName="EmailTemplates/NewAccountTemplate.htm" From="admin@KhanIndustries.co.uk"
       IsBodyHtml="True" Subject="Steps to activate your new account..." Priority="High">
</MailDefinition>
```

Algorithm 7 – Registration E-Mail Setup

These properties include, where to locate the email to send, from whom it should be sent by, the subject of said email, priority level and so forth.

However, to actually send the email, an SMTP service must be setup to allow one to send and receive emails. Setting up an SMTP service can be done through the web.config file. Algorithm 7 will only work once the mail setting in the web.config file is defined. Algorithm 8, below shows the code behind setting up the SMTP service. The network host should firstly be defined, along with any port number, username and password needed to connect to this service.

```
<mailSettings>
      <smtp>
        <network host="relay.jangosmtp.net" port="25" userName="***" password="***" />
      </smtp>
</mailSettings>
```

Algorithm 8 – Web.config MailSettings

Algorithm 9, describes the sending of the mail, it firstly gets the userid of the recently registered user, the algorithm then proceeds to creates the url string which the user must click to authenticate there account.

```
public void CreateUserWizard1_SendingMail(object sender, System.Web.UI.WebControls.MailMessageEventArgs e)
{
            MembershipUser newUser = Membership.GetUser(CreateUserWizard1.UserName);
            Guid newUserId = (Guid)newUser.ProviderUserKey;


            string urlBase = Request.Url.GetLeftPart(UriPartial.Authority) + Request.ApplicationPath;
            string verifyUrl = "Verify.aspx?ID=" + newUserId.ToString();
            string fullUrl = urlBase + verifyUrl;


            e.Message.Body = e.Message.Body.Replace("<%VerificationUrl%>", fullUrl);
}
```

Algorithm 9 – Registration onSendingMail

This url string is then inserted into the body of the file email being sent, which is presented in algorithm 7. Below, you can see the entire email, which the user will have sent to his email.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Steps to activate your account...</title>
</head>
<body style="font-family: Verdana;">
    <h2>
        Welcome to My Website!</h2>
    <p>
        Hello,
        <%UserName%>. You are receiving this email because you recently created a new account
        at my site. Before you can login, however, you need to first visit the following
        link:</p>
    <p><%VerificationUrl%></p>
    <p>After visiting the above link you can log into the site!</p>
    <p>If you have any problems verifying your account, please reply to this email to get assistance.</p>
    <p>Thanks!</p>
</body>
</html>
```

Algorithm 10 – Registration Authentication Email

The final post-condition for registration function is that a UserProfile Row must be inserted into the table UserProfile, representing the user, whom had just registered. So that when they do login, under profile there username details will be there and other extra details the user can fill in. This is done with the second property OnCreatedUser which can be found in Algorithm 5.

```
protected void CreateUserWizard1_CreatedUser(object sender, EventArgs e)
{
            MembershipUser newUser = Membership.GetUser(CreateUserWizard1.UserName);
            Guid newUserId = (Guid)newUser.ProviderUserKey;


            string connectionString =
ConfigurationManager.ConnectionStrings["HotelProjectConnectionString"].ConnectionString;
            string insertSql = "INSERT INTO UserProfile(UserId, FirstName, LastName, PhoneNo, HouseNo,
StreetName, City, Country, PostCode, Email) VALUES(@UserId, @FirstName, @LastName, @PhoneNo, @HouseNo,
@StreetName, @City, @Country, @PostCode, @Email)";

            using (SqlConnection myConnection = new SqlConnection(connectionString))
            {
                myConnection.Open();

                SqlCommand myCommand = new SqlCommand(insertSql, myConnection);
                myCommand.Parameters.AddWithValue("@UserId", newUserId);
                myCommand.Parameters.AddWithValue("@FirstName", DBNull.Value);
                myCommand.Parameters.AddWithValue("@LastName", DBNull.Value);
                myCommand.Parameters.AddWithValue("@PhoneNo", DBNull.Value);
                myCommand.Parameters.AddWithValue("@HouseNo", DBNull.Value);
                myCommand.Parameters.AddWithValue("@StreetName", DBNull.Value);
                myCommand.Parameters.AddWithValue("@City", DBNull.Value);
                myCommand.Parameters.AddWithValue("@Country", DBNull.Value);
                myCommand.Parameters.AddWithValue("@PostCode", DBNull.Value);
                myCommand.Parameters.AddWithValue("@Email", DBNull.Value);
                myCommand.ExecuteNonQuery();

                myConnection.Close();
            }
        }
```

Algorithm 11 – Registration Inserted User Profile Upon User Creation

Algorithm 11, gets the unique userid created, it then establishes a connection with the database, defines the properties for the insert sql, opens the connection to the databse, inserts the parameter guid newUserID into the first parameter of the UserProfile row UserID , and leaves the rest null for the user to enter. Once this is done the connection is closed off to the database. This concludes the functionality of the registration system, ensuring the pre and post conditions are initialised.

4.2.4.2 Login

A login function has also been identified in the requirements, having no real pre or post condition other than that a user must be already registered to login and validation must be present.

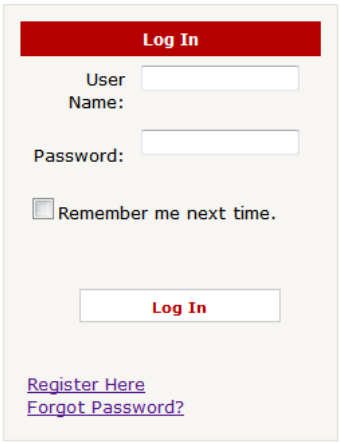


Figure 53 – Login Function

```
<asp:Login ID="Login1" runat="server" DestinationPageUrl="~/Default.aspx"
        CreateUserText="Register Here"
        CreateUserUrl="Registration.aspx
        PasswordRecoveryText="Forgot Password?"
        PasswordRecoveryUrl="~/RecoverPassword.aspx">
</asp:Login>
```

Algorithm 12 – Login Control

The Login Control as shown in Algorithm 12, will define a destination page the user will be sent to upon successful login. Furthermore, extra properties are also defined, such a link to the registration page if they have not registered already and a password recovery link.

```
<asp:RequiredFieldValidator ID="PasswordRequired" runat="server"
       ControlToValidate="Password" ErrorMessage="Password is required."
       ToolTip="Password is required." ValidationGroup="Login1">
*</asp:RequiredFieldValidator>
```

Algorithm 13 – Login Validation

As defined in Algorithm 13, similar validation to the registration function will be used for the login function. However, the validation for this is slightly different as asp automatically will query the aspnetdb tables that have been inserted into the hotel database. To see if the user credentials inputted match any user present in the hotel database. If the details could not be found, an error message will be sent back as shown in Algorithm 14 below.

```
<asp:Literal ID="FailureText" runat="server" EnableViewState="False"></asp:Literal>
```

Algorithm 14 – Login Failure

4.2.4.3 Forgot Password

A functional requirement allowing a user to recover their password if they have forgotten it should be provided. The pre-condition is that the user must be a registered user, as well as entering in the username. The post condition, is that upon requesting recovery, an email should be sent to the user with a new password.

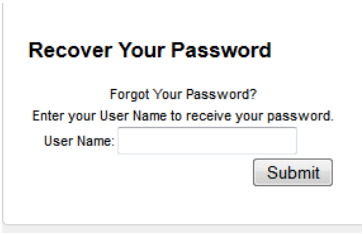


Figure 54 – Forgot Password Control

```
<asp:PasswordRecovery ID="RecoverPwd" runat="server" onsendingmail="RecoverPwd_SendingMail">
     <MailDefinition BodyFileName="EmailTemplates/PasswordRecovery.txt" Subject="Your password has been
reset..."  From="admin@KhanIndustries.co.uk" Priority="High">
      </MailDefinition>
</asp:PasswordRecovery>
```

Algorithm 15 – Forgot Password Control

Algorithm 15 defines the password recovery control, including the mail definition, which describes the properties of the email, to then send to the user upon clicking the “Recover Password” button.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Steps to Recover Password...</title>
</head>
<body style="font-family: Verdana;">
Your password has been reset, <%UserName%>!

According to our records, you have requested that your password be reset. Your new password is: <%Password%>

If you have any further questions or trouble logging on please contact a site administrator.

Thank you!
```

Algorithm 16 – Forgot Password Mail

The recovery password control will automatically generate a temporary password and replace it with <%password%> shown in the body of this email. The user is then capable of logging into their account using their existing username and the new password. They will then be able to change their password once logged in, we will be looking at this functionality later.

4.2.4.4 Update Profile

Registered users should be allowed to change any of the profile details once they are logged into the system. The Pre-Condition of this functionality is that the user must one be already a registered member and two the user must first be logged in to access their account section. Also the post-condition for this functionality is that once the user changes his details and updates, these changes should be directly made to the database.

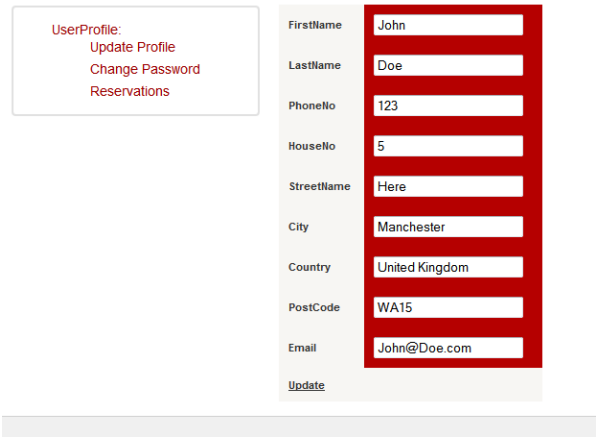


Figure 55 – Update Profile Control

```
<asp:DetailsView ID="UserProfile" runat="server" AutoGenerateRows="False" DataKeyNames="UserId"
            DataSourceID="UserProfileDatabase" DefaultMode="Edit"
            OnItemUpdated="UserProfile_ItemUpdated"
            GridLines="None">
            <Fields>
                <asp:BoundField DataField="FirstName" HeaderText="FirstName" SortExpression="FirstName"/>
                <asp:BoundField DataField="LastName" HeaderText="LastName" SortExpression="LastName"/>
                <asp:BoundField DataField="PhoneNo" HeaderText="PhoneNo" SortExpression="PhoneNo"/>
                <asp:BoundField DataField="HouseNo" HeaderText="HouseNo" SortExpression="HouseNo"/>
                <asp:BoundField DataField="StreetName" HeaderText="StreetName" SortExpression="StreetName"/>
                <asp:BoundField DataField="City" HeaderText="City" SortExpression="City"/>
                <asp:BoundField DataField="Country" HeaderText="Country" SortExpression="Country"/>
                <asp:BoundField DataField="PostCode" HeaderText="PostCode" SortExpression="PostCode"/>
                <asp:BoundField DataField="Email" HeaderText="Email" SortExpression="Email"/>
                <asp:CommandField ShowEditButton="True" ShowCancelButton="False" />
            </Fields>
</asp:DetailsView>
```

Algorithm 17 – Update Profile Control Details View

Using the asp details view control we are able to show the details of the user in a certain way as displayed in figure 55. You can see this asp.net code for this shown in Algorithm 17. Algorithm 17 shows that the user profile is generated based on the DataKeyName “UserID”, once the UserID is found, the rest of the user details can be queried from the database. The UserID is derived from the C# code in the UpdateProfile.aspx.cs file.

```
protected void UserProfileDataSource_Selecting(object sender,SqlDataSourceSelectingEventArgs e)
{

            // Get a reference to the currently logged on user
            MembershipUser currentUser = Membership.GetUser();

             // Determine the currently logged on user's UserId value
            Guid currentUserId = (Guid)currentUser.ProviderUserKey;

             // Assign the currently logged on user's UserId to the @UserId parameter
             e.Command.Parameters["@UserId"].Value = currentUserId;

}
```

Algorithm 18 – Update Profile Get Logged On UserID

Algorithm 18, defines the current user logged in, it does this by putting the currently logged on user into a Membership User Object. Once this has been done we can extract the UserID i.e. the GUID from the currently logged in user. This is an important first step to achieving the functionality of updating the user profile, as before we can get any other information of the user this GUID must be determined of each user, who is currently logged in.

Furthermore, as you can see in Algorithm 17 a data source is defined, in this case the data source is point to the UserProfile tale in the hotel Database. Also to note, the default mode is set to edit so that is made easier for the user to their details.

```
<asp:SqlDataSource ID="UserProfileDatabase" runat="server"
         ConnectionString="<%$ ConnectionStrings:HotelProjectConnectionString %>"
         SelectCommand="SELECT [UserId], [FirstName], [LastName], [PhoneNo], [HouseNo], [StreetName], [City],
[Country], [PostCode], [Email] FROM [UserProfile] WHERE ([UserId] = @UserId)"
         OnSelecting="UserProfileDataSource_Selecting"
         UpdateCommand="UPDATE UserProfile SET FirstName = @FirstName, LastName = @LastName, PhoneNo = @PhoneNo
         , HouseNo = @HouseNo, StreetName = @StreetName, City = @City, Country = @Country, PostCode = @PostCode,
         Email = @Email WHERE UserId = @UserId">
            <SelectParameters>
                <asp:Parameter Name="UserId" Type="Object" />
            </SelectParameters>
            <UpdateParameters>
                <asp:Parameter Name="FirstName" />
                <asp:Parameter Name="LastName" />
                <asp:Parameter Name="PhoneNo" />
                <asp:Parameter Name="HouseNo" />
                <asp:Parameter Name="StreeName" />
                <asp:Parameter Name="City" />
                <asp:Parameter Name="Country" />
                <asp:Parameter Name="PostCode" />
                <asp:Parameter Name="Email" />
                <asp:Parameter Name="UserId" />
            </UpdateParameters>
        </asp:SqlDataSource>
```

Algorithm 19 – Update Profile Data Source

Algorithm 19, represents the connection of the Update Profile data source. Firstly, it creates the connect ion to the database using the connect string which is defined in the web.config file. By using the Select command, it allows the system to retrieve any existing user information, which may be stored already in the database. If the user changes any of the details, the update command will query the database, so the new information is directly stored on the database. The system recognizes which information should be stored in which column of the database, as the parameters In the data source have been assigned to the same value of the bound field inputs in the details view found in Algorithm 17.

As soon as a user submits any update, the itemUpdateProperty is activated. The function enables a hidden asp label by setting its visible property to the. This label makes it clear to the user that they have updated there profile details. This is shown below in Algorithm 20.

```
protected void UserProfile_ItemUpdated(object sender, DetailsViewUpdatedEventArgs e)
{
            SettingsMessage.Visible = true;
}

<asp:Label ID="SettingsMessage" runat="server" Text="Your settings have been updated."
            EnableViewState="false" Visible="false"></asp:Label>
```

Algorithm 20 – Update Profile Confirmation

4.2.4.5 Change Password

The ability to change a user's password is another functional requirement defined in the analysis and design chapter. The Pre-condition is that the user must first be logged in. The post condition is that an email will be sent to the user confirming the password change.

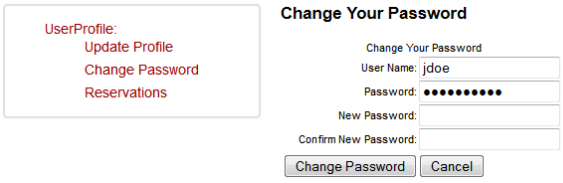


Figure 56 – Change Password Control

```
<asp:ChangePassword ID="ChangePwd" runat="server" ContinueDestinationPageUrl="~/UserProfile/UpdateProfile.aspx"
    DisplayUserName="True">
    <MailDefinition BodyFileName="~/EmailTemplates/ChangePassword.htm"
       IsBodyHtml="True" Subject="Your password has been changed!" From="kkhan1991@hotmail.com" Priority="High">
    </MailDefinition>
</asp:ChangePassword>
```

Algorithm 21 – Change Password Control

Algorithm 21, initialises and defines the Change Password Control, this function is very similar to the recover password function. To change a user's password, they must first be able to provide their username and current password, they should then provide a new password.

Once they have changed their password, an email is sent to the user confirming them of this, thus fulfilling the post condition of this function. The email is defined in the mail definition of this function. The email can be found in Algorithm 22 below.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<h2>Your Password Has Been Changed!</h2>
    <p>
        This email confirms that your password has been changed.
    </p>
    <p>
        To log on to the site, use the following credentials:
    </p>
    <table>
        <tr>
            <td>
                <b>Username:</b>
            </td>
            <td>
                <%UserName%>
            </td>
        </tr>
        <tr>
            <td>
                <b>Password:</b>
            </td>
            <td>
                <%Password%>
            </td>
        </tr>
    </table>
    <p>
        If you have any questions or encounter any problems logging in,
```

Algorithm 22 – Change Password Email

4.2.4.6 Cancel Reservations

Cancel Reservation is higher priority function than some others covered so far. Users should be allowed to cancel their current reservation through their account page, once they have logged in. The Pre-condition of this function is that the user must have an existing reservation. The post condition is that the changes should be made directly to the database.

```
<asp:GridView ID="cancelReservation" runat="server" AutoGenerateRows="False" AutoGenerateColumns="False"
        DataKeyNames="ReservationID" DataSourceID="SqlDataSource1">
        <Columns>
            <asp:CommandField ShowDeleteButton="True" />
            <asp:BoundField DataField="ReservationID" HeaderText="ReservationID"
                InsertVisible="False" ReadOnly="True" SortExpression="ReservationID" />
            <asp:BoundField DataField="RoomNo" HeaderText="RoomNo"
                SortExpression="RoomNo" />
            <asp:BoundField DataField="CheckInDate" HeaderText="CheckInDate"
                SortExpression="CheckInDate" />
            <asp:BoundField DataField="CheckOutDate" HeaderText="CheckOutDate"
                SortExpression="CheckOutDate" />
            <asp:BoundField DataField="NoOfDays" HeaderText="NoOfDays"
                SortExpression="NoOfDays" />
        </Columns>
</asp:GridView>
```

Algorithm 23 – Cancel Reservation Control

Algorithm 23, represents the code for defining the Data Grid View for showing the users Reservation. Like seen in previous functions, in order to populate such Grid View, a data source must be defined. Also to note that for this GridView a commandfield has been appended to it called ShowDeleteButton. This buttons allows a user to efficiently delete/cancel a reservation. This change is made directly to the database.

```
<asp:SqlDataSource ID="cancelReservationDataSource" runat="server"
        ConnectionString="<%$ ConnectionStrings:HotelProjectConnectionString %>"

         SelectCommand="SELECT [ReservationID], [UserId], [RoomNo], [CheckInDate], [CheckOutDate], [NoOfDays]
FROM [Reservations] WHERE ([UserId] = @UserId)" OnSelecting = "SqlData_Selecting"

        DeleteCommand="DELETE FROM [Reservations] WHERE [ReservationID] = @ReservationID"
            <SelectParameters>
                <asp:Parameter Name="UserId" Type="Object" />
            </SelectParameters>
            <DeleteParameters>
                <asp:Parameter Name="ReservationID" Type="Int32" />
            </DeleteParameters>
</asp:SqlDataSource>
```

Algorithm 24 – Cancel Reservation Data Source

Algorithm 24 as shown above, defines the data source for the grid view defined in Algorithm 23. The code sets the connection to the database. Like shown in previous functions the system uses an SQL command to select all the reservation that the user may currently have, it does this by taking in the

UserID of the current logged in user. In addition, using the Sql Delete the system is able to delete the reservation based on the reservation ID provided in the parameter.

4.2.4.7 Search

As mentioned in the Literature Review in Chapter 2. A search function is used in all example system reviewed. Most searches have been made easy to use by providing search boxes for Check in and Check out dates. The System uses Ajax to achieve this.

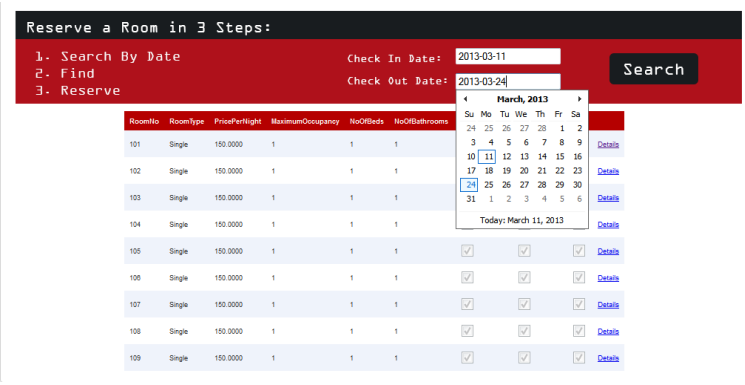


Figure 57 – Search Function

Figure 57, represents the search page and function, which consists of two text boxes and a search button. As mentioned above the textbox's have an Ajax Calendar Extender to make it easier for the user to pick a certain date in a certain format.

```
<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>
```

Algorithm 25 – Enabling Ajax

Before we can add any Ajax tools to our system, we first need to enable Ajax. This is done in algorithm 25.

```
<asp:ScriptManager ID="ScriptManager1" runat="server">
    </asp:ScriptManager>

<asp:TextBox ID="txtboxCheckInDate" runat="server"></asp:TextBox>
  <asp:CalendarExtender ID="CalendarExtender1" runat="server" TargetControlID="txtboxCheckInDate" Format="yyyy-
MM-dd">
  </asp:CalendarExtender>
```

Algorithm 26 – Search Ajax Calendar

Algorithm 26, shows the enabling of the script manager, so that the system is able to use some of the controls in the Ajax tool kit. Once this has been initialised, the Calendar Extender. The property "TargetControlID" tell the Ajax Extender on what asp:control the calendar should be enabled on, in this case it is the CheckIn Textbox. Extra properties can also be specified such as the format of the date received from selecting a date. This is especially useful as the system can set the format of the date to the same format of the date stored in our database.

Upon clicking the search button, the search page will post back and automatically generate a data grid view based on the parameters of search inputs the user has entered.

```
<asp:GridView ID="gvSearch" ShowHeaderWhenEmpty="true" ShowHeader="true" runat="server"
AutoGenerateColumns="False" DataKeyNames="RoomNo"
DataSourceID="SearchDataSource"/>
   <Columns>
                <asp:BoundField DataField="RoomNo" HeaderText="RoomNo" ReadOnly="True" SortExpression="RoomNo"
/>
                <asp:BoundField DataField="RoomType" HeaderText="RoomType" SortExpression="RoomType" />
                <asp:BoundField DataField="PricePerNight" HeaderText="PricePerNight"
SortExpression="PricePerNight" />
                <asp:BoundField DataField="MaximumOccupancy" HeaderText="MaximumOccupancy"
SortExpression="MaximumOccupancy" />
                <asp:BoundField DataField="NoOfBeds" HeaderText="NoOfBeds" SortExpression="NoOfBeds" />
                <asp:BoundField DataField="NoOfBathrooms" HeaderText="NoOfBathrooms"
SortExpression="NoOfBathrooms" />
                <asp:CheckBoxField DataField="Entertainment" HeaderText="Entertainment"
SortExpression="Entertainment" />
                <asp:CheckBoxField DataField="RoomService" HeaderText="RoomService" SortExpression="RoomService"
/>
                <asp:CheckBoxField DataField="Gym" HeaderText="Gym" SortExpression="Gym" />
                <asp:HyperLinkField Text="Details" DataNavigateUrlFields="RoomNo"
                    DataNavigateUrlFormatString="SearchDetails.aspx?id={0}" />
            </Columns>
        </asp:GridView>
```

Algorithm 27 – Search DataGridView

Algorithm 27 displayed the contents of the data grid view for the search results acquired. This grid view like shown in other functions, gets populated with a connection to a data source. In this data grid view there is an column, defined as Details. This column is a link to a detailed page of the specific room that the user click. They are linked to this room through the url string "SearchDetails.aspx?id={0} where 0 is room ID.

```
<asp:SqlDataSource ID="SearchDataSource" runat="server" ConnectionString="<%$
ConnectionStrings:HotelProjectConnectionString %>"
            SelectCommand="SELECT * FROM Rooms WHERE NOT EXISTS(SELECT * FROM Reservations WHERE Rooms.RoomNo =
Reservations.RoomNo AND @CheckIn &lt;= Reservations.CheckOutDate AND @CheckOut &gt;= Reservations.CheckInDate)">
            <SelectParameters>
                <asp:ControlParameter ControlID="txtboxCheckInDate" Name="CheckIn" PropertyName="Text" />
                <asp:ControlParameter ControlID="txtboxCheckOutDate" Name="CheckOut" PropertyName="Text" />
            </SelectParameters>
</asp:SqlDataSource>
```

Algorithm 28 – Search Data Source

Algorithm 28, shows the datasource for the search function, note the difference in this data source to others developed before. This data source queries a more complex SQL statement. The query defines all room where there is currently no reservation between the two dates the user has entered. We use control parameters to tell the query what the two dates entered are.

4.2.4.8 Book Reservation

Once a user has searched for a room and are on the details page, they should be allowed to a book the room for reservation. The precondition for booking such room is that they should be logged in. The post condition is that the reservation should be stored directly to the database.

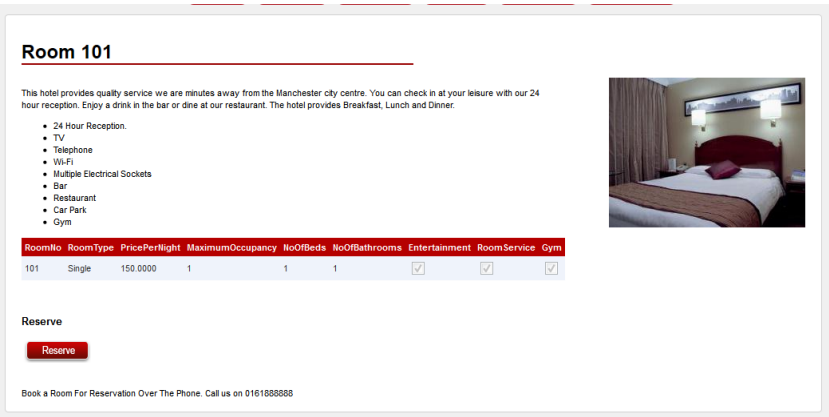


Figure 58 – Search Details Page

Figure 58, represents the typical display of the details view of a specified room. Note that there is a reservation button this page allowing users to reserve the room for booking that they are on, based on the check in and check out date specified earlier.

The reserve button must get the user currently logged in with the check in and check out dates and store this reservation data inside the database. However, if a user is not logged in an appropriate error message must be displayed.

```
  <asp:ImageButton ID="ReserveButton" runat="server" onclick="ReserveButton_Click"
ImageUrl="~/Images/btn_reserve_normal.png" CssClass="ReserveButton"/>
```

Algorithm 29 – Reserve Button

```
        protected void ReserveButton_Click(object sender, ImageClickEventArgs e)
        {

            try
            {
                // Get a reference to the currently logged on user
                MembershipUser currentUser = Membership.GetUser();

                // Determine the currently logged on user's UserId value
                Guid currentUserId = (Guid)currentUser.ProviderUserKey;



                string strRoomNo = Request.QueryString["Id"].ToString();
                SqlConnection sqlCon = new
SqlConnection(ConfigurationManager.ConnectionStrings["HotelProjectConnectionString"].ConnectionString);
                sqlCon.Open();
                SqlCommand insert = new SqlCommand("INSERT
Reservations([UserID],[RoomNo],[CheckInDate],[CheckOutDate]) VALUES (@UserID, @RoomNo, @CheckInDate,
@CheckOutDate)", sqlCon);
                //insert.Parameters["@UserID"].Value = currentUserId;
                insert.Parameters.AddWithValue("@UserID", currentUserId);
                insert.Parameters.AddWithValue("@RoomNo", strRoomNo);
                insert.Parameters.AddWithValue("@CheckInDate", GlobalVariable.GlobalCheckInDate);
                insert.Parameters.AddWithValue("@CheckOutDate", GlobalVariable.GlobalCheckOutDate);

                insert.ExecuteReader();

                sqlCon.Close();

                Response.Redirect("~/SucessfullReserve.aspx");
            }
            catch
            {
                Label1.Text = "You Must Be Logged In To Reserve A Room";
            }

        }
```

Algorithm 30 – Reserve Room

Algorithm 30, shows the code used to reserve the room, it first determines whether a user is logged in or not, if they are not, an appropriate error message is sent back and displayed to them. If however, there is someone logged in, it proceeds to opening the connection to the database and inserting a row into the reservation table. With the details of the roomNo, check-in and checkout date. Once this is done the connection is closed and the user is directed to a page, confirming the reservation.

4.2.5 Admin Controls

4.2.5.1 Manage Users

The management of users is not a requirement listed in chapter 3, however, this was something vital to have for a management information system. It also makes it easier for admins to manage their users, such as approving and/or unlocking their account.

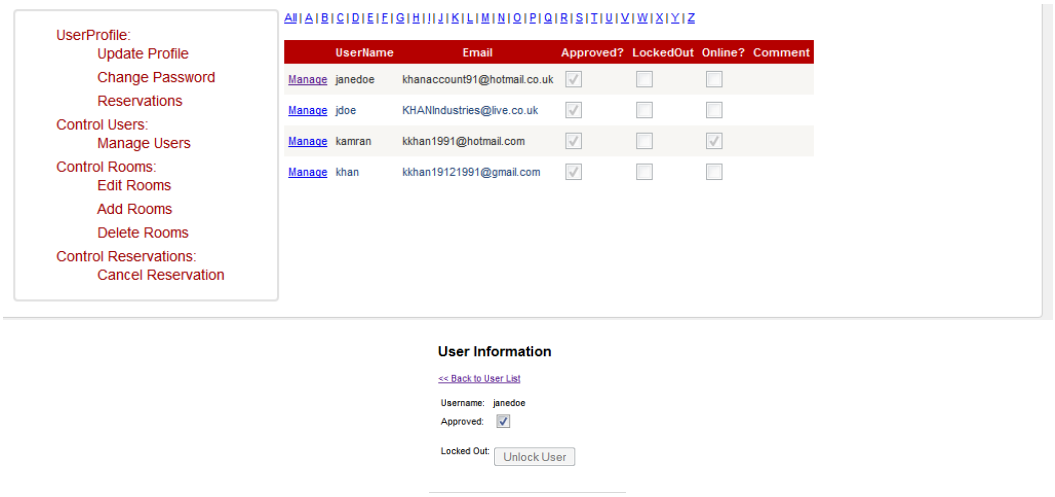


Figure 59 – Manage Users

Figure 59, shows the manage users page, essentially listing all users with their username, email, approved status, locked out status and online status. Functionality, to allow an admin to approve a person to login has been given as well as the ability unlock a user, If they have been locked out due to entering a password in wrongly, too many times.

```
protected void Page_Load(object sender, EventArgs e)
        {
            if (!Page.IsPostBack)
            {

                // If querystring value is missing, send the user to ManageUsers.aspx
                string userName = Request.QueryString["user"];
                if (string.IsNullOrEmpty(userName))
                    Response.Redirect("ManageUsers.aspx");

                // Get information about this user
                MembershipUser usr = Membership.GetUser(userName);
                if (usr == null)
                    Response.Redirect("ManageUsers.aspx");

                UserNameLabel.Text = usr.UserName;
                IsApproved.Checked = usr.IsApproved;
                if (usr.LastLockoutDate.Year < 2000)

                    LastLockoutDateLabel.Text = string.Empty;
                else
                    LastLockoutDateLabel.Text = usr.LastLockoutDate.ToShortDateString();

                UnlockUserButton.Enabled = usr.IsLockedOut;
            }

        }
```

Algorithm 31 – Manage User / User Detail Information

When an admin choses to further manage a user, the user information page is loaded, which provides some brief information and data to change. Algorithm 31 describes the process of showing that information. Firstly, the username of the person taken from the previous page is retrieved, there approval status is shown and also a button showing whether they are unlocked or locked is displayed.

```
protected void IsApproved_CheckedChanged(object sender, EventArgs e)
{
            // Toggle the user's approved status
            string userName = Request.QueryString["user"];
            MembershipUser usr = Membership.GetUser(userName);
            usr.IsApproved = IsApproved.Checked;
            Membership.UpdateUser(usr);
}
```

Algorithm 32 – Manage Users Approval Status

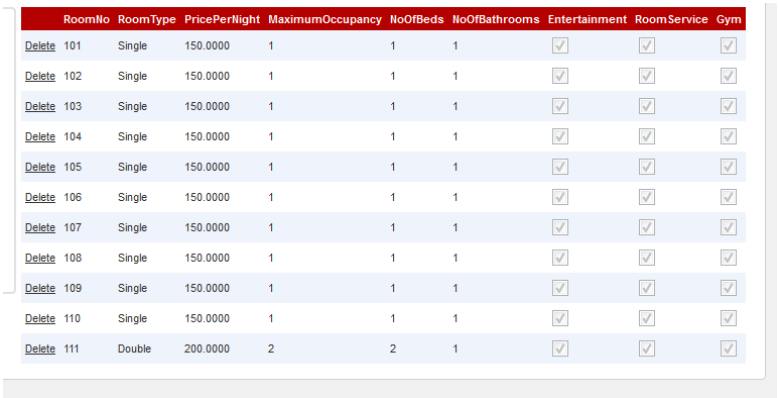
```
protected void UnlockUserButton_Click(object sender, EventArgs e)
{
            // Unlock the user account
            string userName = Request.QueryString["user"];
            MembershipUser usr = Membership.GetUser(userName);

            usr.UnlockUser();
            UnlockUserButton.Enabled = false;
            StatusMessage.Text = "The user account has been unlocked.";
}
```

Algorithm 33 – Manage Users Locked Status

Algorthm 32, shows the code to allow the admin to change the approved status of a specific user. Whereas algorithm 33 shows the code to allow the admin to change the locked status of a user. Once these statuses are changed an appropriate feedback message is displayed.

4.2.5.2 Manage Rooms

Being able to manage rooms is a key requirement factor that must be implemented in order to call this information system a management information system for hotels. Being able to manage rooms, should mean that an admin can use the website to carry out functions such as adding a new room, editing a room and deleting a room.



	RoomNo	RoomType	PricePerNight	MaximumOccupancy	NoOfBeds	NoOfBathrooms	Entertainment	RoomService	Gym
Delete	101	Single	150.0000	1	1	1	☑	☑	☑
Delete	102	Single	150.0000	1	1	1	☑	☑	☑
Delete	103	Single	150.0000	1	1	1	☑	☑	☑
Delete	104	Single	150.0000	1	1	1	☑	☑	☑
Delete	105	Single	150.0000	1	1	1	☑	☑	☑
Delete	106	Single	150.0000	1	1	1	☑	☑	☑
Delete	107	Single	150.0000	1	1	1	☑	☑	☑
Delete	108	Single	150.0000	1	1	1	☑	☑	☑
Delete	109	Single	150.0000	1	1	1	☑	☑	☑
Delete	110	Single	150.0000	1	1	1	☑	☑	☑
Delete	111	Double	200.0000	2	2	1	☑	☑	☑

Figure 60 – Manage Rooms Delete

Figure 60 shows the information presented to the admin to allow them to delete any room, like shown in previous parts of the development chapter the data grid is a very good tool for dynamic data driven websites. An external column (delete) which is not stored in the database has been added to this datagridview to allow the admin to delete a room with a single click.

```
<asp:GridView ID="DeleteRooms" runat="server" AutoGenerateColumns="False"
        CellPadding="4" DataKeyNames="RoomNo" DataSourceID="DeleteRooms"/>
        <Columns>
            <asp:CommandField ShowDeleteButton="True" />
            <asp:BoundField DataField="RoomNo" HeaderText="RoomNo" ReadOnly="True"
                SortExpression="RoomNo" />
            <asp:BoundField DataField="RoomType" HeaderText="RoomType"
                SortExpression="RoomType" />
            <asp:BoundField DataField="PricePerNight" HeaderText="PricePerNight"
                SortExpression="PricePerNight" />
            <asp:BoundField DataField="MaximumOccupancy" HeaderText="MaximumOccupancy"
                SortExpression="MaximumOccupancy" />
            <asp:BoundField DataField="NoOfBeds" HeaderText="NoOfBeds"
                SortExpression="NoOfBeds" />
            <asp:BoundField DataField="NoOfBathrooms" HeaderText="NoOfBathrooms"
                SortExpression="NoOfBathrooms" />
            <asp:CheckBoxField DataField="Entertainment" HeaderText="Entertainment"
                SortExpression="Entertainment" />
            <asp:CheckBoxField DataField="RoomService" HeaderText="RoomService"
                SortExpression="RoomService" />
            <asp:CheckBoxField DataField="Gym" HeaderText="Gym" SortExpression="Gym" />
        </Columns>
</asp:GridView>
```

Algorithm 34 – Manage Rooms Delete DataGrid

Algorithm 34, shows the datagrid code for populating the rooms. As previously mentioned in dynamic data driven websites the process of changing information such as deleting gets done directly to the database.

```
<asp:SqlDataSource ID="DeleteRoom" runat="server"
        ConnectionString="<%$ ConnectionStrings:HotelProjectConnectionString %>"
        DeleteCommand="DELETE FROM [Rooms] WHERE [RoomNo] = @RoomNo"
        SelectCommand="SELECT * FROM [Rooms]">
        <DeleteParameters>
            <asp:Parameter Name="RoomNo" Type="Int32" />
</asp:SqlDataSource>
```

Algorithm 35 – Manage Rooms Delete SQL Data Source

Algorithm 35 shows the data source and the sql queries .The system first selects all of the rooms stored in the database and then uses another query to delete a specified room. The room being deleted will be the room which the user selects this is shown in the sql statement with the where clause pointing to the parameter @RoomNo which is a data field in the datagrid.

Figure 61 – Manage Rooms Edit

Figure 61 represents the page shown to admin for editing rooms, it shows the datagrid view of all rooms currently in the database. By clicking the edit link they are able to change the information of these rooms.

```
<asp:GridView ID="dgvRooms" runat="server" AllowPaging="True"
     AllowSorting="True" AutoGenerateColumns="False" CellPadding="4"
     DataKeyNames="RoomNo" DataSourceID="RoomDataSource”/>
     <Columns>
         <asp:HyperLinkField DataNavigateUrlFields="RoomNo"
             DataNavigateUrlFormatString="ManageRoomDetails.aspx?id={0}" Text="Edit" />
         <asp:BoundField DataField="RoomNo" HeaderText="RoomNo" ReadOnly="True"
             SortExpression="RoomNo" />
         <asp:BoundField DataField="RoomType" HeaderText="RoomType"
             SortExpression="RoomType" />
         <asp:BoundField DataField="PricePerNight" HeaderText="PricePerNight"
             SortExpression="PricePerNight" />
         <asp:BoundField DataField="MaximumOccupancy" HeaderText="MaximumOccupancy"
             SortExpression="MaximumOccupancy" />
         <asp:BoundField DataField="NoOfBeds" HeaderText="NoOfBeds"
             SortExpression="NoOfBeds" />
         <asp:BoundField DataField="NoOfBathrooms" HeaderText="NoOfBathrooms"
             SortExpression="NoOfBathrooms" />
         <asp:CheckBoxField DataField="Entertainment" HeaderText="Entertainment"
             SortExpression="Entertainment" />
         <asp:CheckBoxField DataField="RoomService" HeaderText="RoomService"
             SortExpression="RoomService" />
         <asp:CheckBoxField DataField="Gym" HeaderText="Gym" SortExpression="Gym" />
     </Columns>
 </asp:GridView>
```

Algorithm 36 - Manage Rooms Edit

Algorithm 36 shows the code for the datagridview of edit rooms. The extra columns “edit” is also inserted here. This column has extra properties so that upon clicking, the admin is taken to a further page to actually edit the information of the specified room.

```
    <asp:SqlDataSource ID="RoomDataSource" runat="server"
        ConnectionString="<%$ ConnectionStrings:HotelProjectConnectionString %>"
        SelectCommand="SELECT [RoomNo], [RoomType], [PricePerNight], [MaximumOccupancy], [NoOfBeds],
[NoOfBathrooms], [Entertainment], [RoomService], [Gym] FROM [Rooms]">
    </asp:SqlDataSource>
```

Algorithm 37 – Manage Rooms Edit Data Source

Algorithm 37 shows the code for the data source, which populates the information in the data grid in algorithm 36.

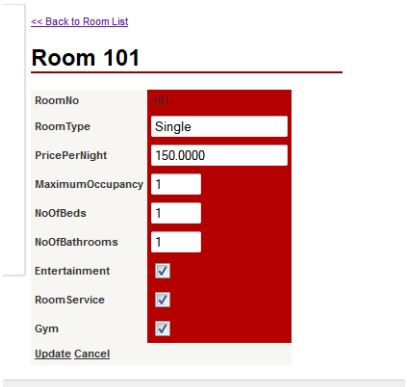


Figure 62 – Manage Rooms Edit Details

Figure 62, shows information presented to an admin once they click the edit button on the datagrid view, they are taken to this page where they can directly change the details of each and every room.

```
    <asp:DetailsView ID="dvRoomDetail" runat="server" Height="50px" Width="125px"
        AutoGenerateEditButton="True" DefaultMode="Edit" AutoGenerateRows="False"
        DataKeyNames="RoomNo" DataSourceID="SqlDataSource1"
        OnItemUpdated="Rooms_ItemUpdated"/>
        <Fields>
            <asp:BoundField DataField="RoomNo" HeaderText="RoomNo" ReadOnly="True"
                SortExpression="RoomNo" />
            <asp:BoundField DataField="RoomType" HeaderText="RoomType"
                SortExpression="RoomType" />
            <asp:BoundField DataField="PricePerNight" HeaderText="PricePerNight"
                SortExpression="PricePerNight" />
            <asp:BoundField DataField="MaximumOccupancy" HeaderText="MaximumOccupancy"
                SortExpression="MaximumOccupancy" />
            <asp:BoundField DataField="NoOfBeds" HeaderText="NoOfBeds"
                SortExpression="NoOfBeds" />
            <asp:BoundField DataField="NoOfBathrooms" HeaderText="NoOfBathrooms"
                SortExpression="NoOfBathrooms" />
            <asp:CheckBoxField DataField="Entertainment" HeaderText="Entertainment"
                SortExpression="Entertainment" />
            <asp:CheckBoxField DataField="RoomService" HeaderText="RoomService"
                SortExpression="RoomService" />
            <asp:CheckBoxField DataField="Gym" HeaderText="Gym" SortExpression="Gym" />
        </Fields>
    </asp:DetailsView>
```

Algorithm 38 – Manage Rooms Edit Details View

Algorithm 38, represents the code used to display the columns of each room and there information. The details view also has an OnItemUpdated event and connects to its own datasource.

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
        ConnectionString="<%$ ConnectionStrings:HotelProjectConnectionString %>"
        SelectCommand="SELECT [RoomNo], [RoomType], [PricePerNight], [MaximumOccupancy], [NoOfBeds],
[NoOfBathrooms], [Entertainment], [RoomService], [Gym] FROM [Rooms] WHERE ([RoomNo] = @RoomNo)"
        UpdateCommand="UPDATE Rooms SET
RoomType=@RoomType,PricePerNight=@PricePerNight,MaximumOccupancy=@MaximumOccupancy,NoOfBeds=@NoOfBeds,NoOfBathro
oms=@NoOfBathrooms,Entertainment=@Entertainment,RoomService=@RoomService,Gym=@Gym WHERE RoomNo=@RoomNo">
        <SelectParameters>
            <asp:ControlParameter ControlID="labelRoomNo" Name="RoomNo" PropertyName="Text"
                Type="Int32" />
        </SelectParameters>
        <UpdateParameters>
                <asp:Parameter Name="RoomType" />
                <asp:Parameter Name="PricePerNight" />
                <asp:Parameter Name="MaximumOccupancy" />
                <asp:Parameter Name="NoOfBeds" />
                <asp:Parameter Name="NoOfBathrooms" />
                <asp:Parameter Name="Entertainment" />
                <asp:Parameter Name="RoomService" />
                <asp:Parameter Name="Gym" />
            </UpdateParameters>
</asp:SqlDataSourc
```

Algorithm 39 – Manage Rooms Edit Details View DataSource

Algorithm 39, represents the sql data source of figure 62. The Sql Select command and update commands are defined along with their parameter's.

```
        protected void Rooms_ItemUpdated(object sender, DetailsViewUpdatedEventArgs e)
        {
            RoomsMessage.Visible = true;
        }
```

Algorithm 40 – Manage Rooms Edit Details ItemUpdate Event

```
<div><asp:Label ID="RoomsMessage" runat="server" Text="Room settings have been updated."
           EnableViewState="false" Visible="false"></asp:Label></div>
```

Algorithm 41 – Manage Rooms Edit Details Update Message

Once the room has been updated with the new details, a message will become visible which is shown in Algorithm 40 and algorithm 41 defines the message to be displayed. This shows a text property displaying to the user that the room settings have been updated.

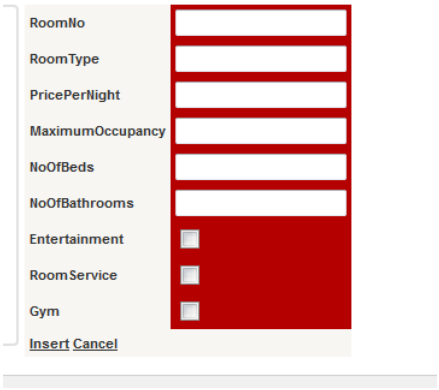


Figure 63 – Manage Rooms Add Rooms

Figure 63 shows the information displayed to the admin on the add rooms page, once there they are able to insert information into the textboxes, these new room details are stored in the database

```
 <asp:DetailsView ID="DetailsView1" runat="server" AutoGenerateRows="False" DefaultMode="Insert"
        CellPadding="4" DataKeyNames="RoomNo" DataSourceID="SqlDataSource1">
        <Fields>
            <asp:BoundField DataField="RoomNo" HeaderText="RoomNo" ReadOnly="False"
                SortExpression="RoomNo" />
            <asp:BoundField DataField="RoomType" HeaderText="RoomType"
                SortExpression="RoomType" />
            <asp:BoundField DataField="PricePerNight" HeaderText="PricePerNight"
                SortExpression="PricePerNight" />
            <asp:BoundField DataField="MaximumOccupancy" HeaderText="MaximumOccupancy"
                SortExpression="MaximumOccupancy" />
            <asp:BoundField DataField="NoOfBeds" HeaderText="NoOfBeds"
                SortExpression="NoOfBeds" />
            <asp:BoundField DataField="NoOfBathrooms" HeaderText="NoOfBathrooms"
                SortExpression="NoOfBathrooms" />
            <asp:CheckBoxField DataField="Entertainment" HeaderText="Entertainment"
                SortExpression="Entertainment" />
            <asp:CheckBoxField DataField="RoomService" HeaderText="RoomService"
                SortExpression="RoomService" />
            <asp:CheckBoxField DataField="Gym" HeaderText="Gym" SortExpression="Gym" />
            <asp:CommandField ShowInsertButton="True" />
        </Fields>
</asp:DetailsView>
```

Algorithm 42 - Manage Rooms Add Rooms Details View

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
        ConnectionString="<%$ ConnectionStrings:HotelProjectConnectionString %>"
        InsertCommand="INSERT INTO [Rooms] ([RoomNo], [RoomType], [PricePerNight], [MaximumOccupancy],
[NoOfBeds], [NoOfBathrooms], [Entertainment], [RoomService], [Gym]) VALUES (@RoomNo, @RoomType,
@PricePerNight, @MaximumOccupancy, @NoOfBeds, @NoOfBathrooms, @Entertainment, @RoomService, @Gym)"
        SelectCommand="SELECT * FROM [Rooms]"
        <InsertParameters>
            <asp:Parameter Name="RoomNo" Type="Int32" />
            <asp:Parameter Name="RoomType" Type="String" />
            <asp:Parameter Name="PricePerNight" Type="Decimal" />
            <asp:Parameter Name="MaximumOccupancy" Type="Int32" />
            <asp:Parameter Name="NoOfBeds" Type="Int32" />
            <asp:Parameter Name="NoOfBathrooms" Type="Int32" />
            <asp:Parameter Name="Entertainment" Type="Boolean" />
            <asp:Parameter Name="RoomService" Type="Boolean" />
            <asp:Parameter Name="Gym" Type="Boolean" />
        </InsertParameters>
</asp:SqlDataSource>
```

Algorithm 43 – Manage Rooms Add Rooms DataSource

Algorithm 42, represents the details view control for displaying the columns and data for the add rooms page. Algorithm 43 represents the data source used to populate the control defined in algorithm 42. The insert command is defined along with the insert parameters which point to the bound fields defined in algorithm 42 for each input field. Thus, completing the management of rooms.

4.2.5.3 Manage Reservations

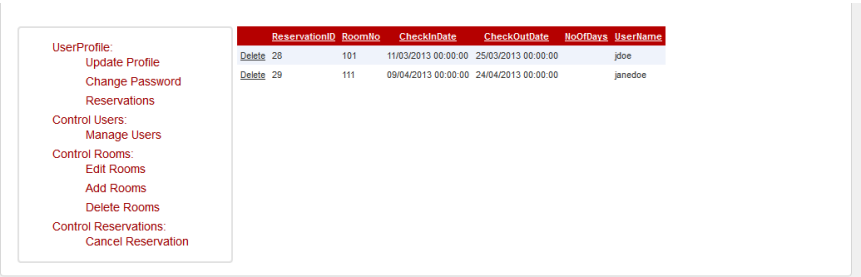


Figure 64 – Manage Reservation Cancel Reservation

Figure 64 shows the information displayed to the admin on the Cancel Reservation page, the page will show all current reservations that every registered user has.

```
<asp:GridView ID="GridView1" runat="server" AllowPaging="True"
        AllowSorting="True" AutoGenerateColumns="False" CellPadding="4"
        DataKeyNames="ReservationID" DataSourceID="SqlDataSource1" ForeColor="#333333"
        GridLines="None">
        <Columns>
            <asp:CommandField ShowDeleteButton="True" />
            <asp:BoundField DataField="ReservationID" HeaderText="ReservationID"
                InsertVisible="False" ReadOnly="True" SortExpression="ReservationID" />
            <asp:BoundField DataField="RoomNo" HeaderText="RoomNo"
                SortExpression="RoomNo" />
            <asp:BoundField DataField="CheckInDate" HeaderText="CheckInDate"
                SortExpression="CheckInDate" />
            <asp:BoundField DataField="CheckOutDate" HeaderText="CheckOutDate"
                SortExpression="CheckOutDate" />
            <asp:BoundField DataField="NoOfDays" HeaderText="NoOfDays"
                SortExpression="NoOfDays" />
            <asp:BoundField DataField="UserName" HeaderText="UserName"
                SortExpression="UserName" />
        </Columns>
</asp:GridView>
```

Algorithm 44 – Manage Reservation Cancel Reservation Data Grid

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
        ConnectionString="<%$ ConnectionStrings:HotelProjectConnectionString %>"
        DeleteCommand="DELETE FROM [Reservations] WHERE [ReservationID] = @ReservationID">
        <DeleteParameters>
            <asp:Parameter Name="ReservationID" Type="Int32" />
        </DeleteParameters>
    </asp:SqlDataSource>
```

Algorithm 45 – Manage Reservation Cancel Reservation Data Source

Algorithm 44, represents the data grid view used to define the columns for the information displayed on the cancel reservation page, whereas algorithm 45 represents the asp and sql code used to populate the information in the data grid view control. This completes the management of reservations

4.3 Summary

In summary this chapter describes the process of implementation with the use of screenshots and code snippets. For every core requirement which was outlined in chapter 3 analysis and design. However, not every line of code for developing the system was given as this was not the purpose for the report. After implementing/developing a product the next stage is Systems Testing. The systems testing will follow on from the development stage in the waterfall model, by giving details of system testing techniques used and their results.

5 System Testing

5.1 Introduction

This chapter will focus on testing stage of the development lifecycle. It is important to test a system after and during the implementation phase so that the system is reliable and free from any errors. As there are many testing techniques one could use, for the purpose of this project, testing techniques reviewed in the literature will be undertaken to test the system. The main testing techniques being used are black box Testing and Website Validation and Compatibility.

5.2 Black Box Testing

Test Case	Test Description	Expected Outcome	Actual Outcome	Results	Evidence Appendix(5)
Registration:					
1.	User tries to Register, whilst leaving some fields blank.	Error Symbol "*" displays, also indicating problems next to certain fields.	Partially Expected	Partially Correct	R1
2.	User tries to register with different matching passwords.	Error Message Displays: The Password and Confirmation Password must match.	As Expected	Correct	R2
3.	Upon Successful Registration	Email verification sent to user.	As Expected	Correct	R3
Login:					
1.	User tries to login with blank fields.	Error Symbol "*" displays, also indicating problems next to certain fields.	Partially Expected	Partially Correct	
2.	User tries to login with incorrect details.	Error Message: Your login attempt was not successful. Please try again.	As Expected	Correct	L2
3.	User Clicks On Forgot Password.	User is taken to forget password page.	As Expected	Correct	
4.	User Clicks On Register Here	User is taken to registration page	As Expected	Correct	
Forgot password:					
1.	User tries to recover password, of a	Error Message: We were unable to access	As Expected	Correct	F1

	username which doesn't exist or empty field..	your information. Please try again.			
2.	Upon Successful Password Recovery	Email sent to user, with a new password.	As Expected	Correct	F2
Update Profile:					
1.	User tries to update Details	Details are updated to database, and message shown confirming update of details.	As Expected	Correct	U1
Change Password:					
1.	Users enters their current password incorrectly.	Error Message: Password incorrect	As Expected	Correct	C1
2.	User leaves some input fields black.	Error Symbol "*" displays, also indicating problems next to certain fields.	Partially Expected	Partially Correct	
Cancel Reservation:					
1.	User tries to cancel their reservation	Reservation cancelled and deleted from database.	As Expected	Correct	CR1
2.	User tries to cancel a previous reservation	Error Message, Unable To Cancel Reservations Which are outdated.	Unexpected, No functionality for this.	Incorrect	
Search For Room:					
1.	User Searches with no dates specified.	Query Brings back empty results.	As Expected	Correct	S1
2.	User Searches with only 1 date specified	Query Brings back empty results	As Expected	Correct	
3.	User Searches with 2 Specified Dates	Query Brings back all rooms which are available between the specified check in and checkout date.	As Expected	Correct	S3
Book Room For Reservation:					
1.	Anonymous User Tries To Book A Room For Reservation	Error Message: You Must Be Logged In To Reserve A Room	As Expected	Correct	B1
2.	Registered User Tries To Book A Room For Reservation	Reservation inserted into database, directed to a confirmation of reservation page.	As Expected	Correct	B2
Manage Rooms:					
1.	Admin tries to add a new room	Rooms gets added to database	As Expected	Correct	
2.	Admin tries to edit a room	Rooms details get updated to database	As Expected	Correct	

3.	Admin tries to delete a room	Room gets deleted from database	As Expected	Correct	
Manage Reservations:					
1.	Admin tries to Cancels any Reservation	Reservation cancelled and removed from database	As Expected	Correct	

Note: not all evidences were provided in the appendix, some functionality is merely repeats of previous ones or too general.

5.3 Website Validation & Compatibility

The web-based information system has been validated and checked using the online w3c markup validation which can be found at (http://validator.w3.org/). This online tool allows one to enter in pieces of code of an entire website address to check the validity of the webpages. W3 validator found 5 errors and 1 warning.

- Errors: 5 required attributes types were not specified.
    - 3 including attribute “alt”.
    - 2 including attribute “type”.

- Warning: No Character Encoding Declared at document level.

These minor validation errors have since been fixed. W3 validator now validate successfully.

Website compatibility is the process of testing the website with the most commonly used online tools for browsing on various different operating systems. (http://browsershots.org). Results can be viewed at (http://browsershots.org/http://www.khanindustries.co.uk/). The Results were that the web-based information system is compatible with the top 3 operating systems including Windows, Mac and Linux. Furthermore, there are several web browsers which the website is compatible in including Mozilla Firefox, Internet Explorer and Google Chrome, Safari, Opera and Iceweasel with various different versions.

No functionality problems occurred during the validation and compatibility checks. Therefore, the objectives for testing outlined in Chapter 1.3 have been successfully met

5.4 Summary

With the use of the testing techniques carried out, many problems initially overlooked have been resolved. Although, there are an infinite number of tests with an infinite number of testing techniques which could be carried out, they cannot all feasibility be conducted. However, the system is in a better position of reliability and error free than it was before carrying out system testing. Evaluating the system under certain criteria's will be looked at in the next chapter.

6 Evaluation

6.1 Introduction

This chapter details the evaluation process conducted on the system. It revisits the objectives which have been set at the beginning of this project, providing information and justification on whether they have been successfully met or not. Furthermore, common heuristics evaluation for a web-based information system is also described. Additionally, for the survey which has been conducted a review and evaluation of this will be given. Finally an evaluation of the methodologies used throughout the project and development will be detailed.

6.2 Evaluation of Objectives

To meet the aim of the overall project, there were several objectives set to help with the constant development of achieving this aim. These objectives provide the evaluation criteria and the level of success, of the project. This section will discuss the objectives set, assessing whether the objectives have been met or not with justification.

Objectives

1. Research techniques and methodologies used to produce an information system, with regard to the Analysis, Design, Development, Testing and Evaluation Stages. Furthermore, review current and previous Hotel Systems.

If one would take a look at the literature review chapter of this report, one would see that every stage of the Systems Development lifecycle has been researched. One would also see with regard to the five stages: Analysis, Design, Development, Testing and Evaluation that the methodologies and techniques used to conduct these stages have also been detailed. Additionally, in the same chapter, a discussion and detailed review has been carried of current and previous Hotel Systems. Therefore, this objective has been successfully met.

2. Analyse the proposed system, documenting the requirements and specifications

Half of chapter 2, the analysis and design chapter contain the analysis work undertaken for this report. If one would view this chapter, there are clear distinct sections of this chapter which are related to the analysis work. For example, there are sections on requirements gathering which detail the process of how the requirements were gathered for this system. Additionally you will see a section detailing the requirements specification for the entire project, which include functional requirements. Furthermore some details outlining any additional requirements such as hardware/software have also been provided. Therefore, this shows this objective has been successfully met.

3. Design the intended system based upon the requirements and specifications documented. Modelling the entire system based upon the techniques derived from the literature review. Additionally create mock-up/wireframe design, for the proposed product.

The second half of chapter 2, the analysis and design chapter. Details the process of designing the proposed system, using system modelling and design techniques detailed and discussed in the literature review chapter. Some of these modelling techniques are specific to certain aspects of the system. For example, the dataflow model is specific, as it only shows the flow of information between business processes/functions, where as other design modelling techniques such as the entity relationship model, shows the relationship between the diff classes of the database and system. Furthermore, towards the end of said chapter mock-up/wireframe designs of the intended web-based system have been produced. Therefore, this objective has been successfully met.

4. Develop the front-end and back-end to the proposed system. It should include all the functional requirements identified.

Chapter 4 of this report has been dedicated to detailing the implementation/development of the system. This chapter details the process taken to implement all of the functional requirements identified, as well as the general non-functional requirements. If one would take a look at the final product/system one would see that all of the functional requirements identified have been implemented, including but not limited to login, registration, search, booking, management functions. Thus this objective has been successfully achieved.

5. Test the system, using test strategies produced and testing systems commonly used, to test a web-based information system.

Chapter 5 of this report is the System Testing chapter. Its sole purpose is to test the core functions of the system, using common test testing technique such as Black box testing. Additionally test methods, such as extra validation and compatibility checks have also been carried out. Therefore this objective has been successfully met.

6.3 Evaluation of Survey

To evaluate the system which has been produced, an anonymous survey was carried out. The volunteers were acquired through a social networking site, where the survey was posted up for anyone to try. The purpose of advertising it on a social networking website was to ensure that a range of volunteers were acquired from people who use the computer regularly and unregularly. As social networking websites tend to bring these two groups of people together. Although the survey concentrated primarily on the functionality and human computer interaction of the system, there were at least 14 questions asked, with opportunity for adding extra comments. Each questions contained a statement about the user interface, the volunteers were then asked to choose from the following:

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

At the end of the survey 10 volunteers took part in it. Producing a diverse range of answers to the questions proposed. The survey was producing using an online tool called survey-monkey. The finding for this survey can be found below.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	Total	Average Rating
The system is aesthetically pleasing	0% 0	0% 0	0% 0	80% 8	20% 2	10	4.20
The system is user friendly	0% 0	0% 0	10% 1	70% 7	20% 2	10	4.10
It is easy to navigate through the system	0% 0	0% 0	10% 1	70% 7	20% 2	10	4.10
Help/Guidance is provided	0% 0	0% 0	70% 7	30% 3	0% 0	10	3.30
There are clear system statuses.	0% 0	0% 0	70% 7	30% 3	0% 0	10	3.30
The system is Flexible	0% 0	0% 0	20% 2	60% 6	20% 2	10	4.00
The system is consistent	0% 0	0% 0	20% 2	40% 4	40% 4	10	4.20

Figure 65 – Human Computer Interaction Questions and Results.

Figure 65, shows the answers of the 10 volunteers for the set of questions regarding Human computer interaction. Although the findings were quite average across the board, the 10 participants agreed mostly on every single statement. As shown above, 70% of the users agreed that there was a neutral level of help/guidance provided. Indicating that more help could be provided when carrying out the functions of the system. Similarly, 70% of them agreed that there wasn't enough clear system status. However, overall the majority of the statements provided the participants agreed with them or strongly agreed with them. Indicating a success regarding the level of human computer interaction provided by the system, but there was still room to improve.

Many of the comments given for this section of the survey described the constructive criticism of the parts represented In the survey above. Most said

"Guidance could be given on carrying out the core functions of the system. Furthermore, there are some system statuses missing, when carrying out certain functionality, such as inserting a new room."

Since carrying out this survey, the statement outlined above with regards to certain functionality has been rectified.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	Total	Average Rating
The system allows for the management of users.	0% 0	0% 0	20% 2	60% 6	20% 2	10	4.00
The system allows for the management of rooms	0% 0	0% 0	10% 1	80% 8	10% 1	10	4.00
The system allows for the management of reservations	0% 0	0% 0	10% 1	80% 8	10% 1	10	4.00
The system provides login/registration functionality	0% 0	0% 0	10% 1	60% 6	30% 3	10	4.20
The system provides a clear way to search for available rooms	0% 0	0% 0	30% 3	50% 5	20% 2	10	3.90
The system provides a way for users to reserve a room.	0% 0	0% 0	20% 2	60% 6	20% 2	10	4.00
The system provides a way to pay for the rooms.	60% 6	0% 0	30% 3	10% 1	0% 0	10	1.90

Figure 66 – Functionality Question and Results

Figure 66 shows the question and results for the functionality side of the survey. Again the 10 participants of the survey mostly agreed upon all the statements. 60% of them agreed that there was no payment system for users to pay for the rooms they have reserved. This is something which will be discussed later in the conclusion chapter under further work which could be done on the system. Furthermore, mostly agreed on the statements of functionality, indicating that majority of functionality was provided in the system, to a more than acceptable standard.

The comments described for this section of the survey mostly said:

"There could be ways of paying for the room using PayPal. Also, more efficient ways for searching a room. i.e. giving users more options to define search criteria, should be provided. "

The statement above will be discussed in the conclusion chapter of this report, regarding further work which could be undertaken to further enhance the system.

Overall the testing has provided good user feedback on the system and ways I can make it more efficient in the short term and also make the system more efficient in the long term with greater functionality.

6.4 Evaluation of Methodologies

The main development methodology used for this project was the waterfall model, as identified in previous chapters. For the majority, the waterfall model has been successful in aiding development. However, elements of agile development with regards to certain aspects of the development lifecycle were implemented when required. Each stage of the system was developed in stages as the waterfall model describes and was completed in a timely fashion according to the Gantt chart. The waterfall model was a good method for development because the requirements for this project were fully understood at the beginning.

6.5 Summary

In this chapter the objectives which have been identified in the first introduction chapter have been reviewed, in terms of determining how successful the objectives have been achieved. Justifications have also been provided for this.

Additionally, an evaluation of the methodologies used with justification has been provided. Including the success of using the waterfall model has also been briefly described.

Furthermore, evaluation was also carried through the use of a survey, from the results gathered during the survey, it was determined that there were some aspects which could be further improved and used for further work.

7 Conclusion

7.1 Introduction

The final chapter of this report, the conclusion chapter, will take an overview of the project as a whole. This chapter takes a look at every stage of the project to summarise the core points. The problems which were met during the time of this project will also be discussed and a brief explanation will be provided on how these problems were overcome. Additionally the author's personal gains and achievements will be looked at. Finally a discussion on how the project could be development further will be given.

7.2 Report Summary

The main body of the report was broken down into 7 chapters including the conclusion chapter. Each chapter is concerned with specific areas of the project, a summary will be provided below of each chapter.

Chapter 1 Introduction, the introductory chapter provided some background information on the project being embarked upon. The aims and objectives were clearly outlined in this chapter as to definitely know what should be achieved.

Chapter 2 Literature Review, the literature chapter laid the foundations for analysing, designing, developing and testing a system such as the one produced for this project. Many different methodologies and techniques for each of the above were also discussed and reviewed. Additionally an analysis was done on past and present systems.

Chapter 3, Systems Analysis and Design, the analysis and design chapter carried out the successful methodologies and techniques which were researched during the literature chapter. This chapter laid the foundations for developing the system because clear requirements and designs were produced.

Chapter 4, Systems Development, the development chapter looked at the core functionally requirements of the system by briefly walking the reader through the process of development for each requirement.

Chapter5, Systems Testing, the testing chapter carried out testing techniques on the system, obtained from the literature to find problems with the system. Some of these problems were initially overlooked. However, in finding issues with the system and then correcting them, it only enhanced the systems efficiency and reliability.

Chapter 6, Systems Evaluation, the evaluation chapter re-identified the objectives set out in the beginning describing if these objectives were successfully met or not. Further evaluation was carried out through the use of surveys to gather feedback from users. Finally an evaluation of methodologies and techniques used were briefly discussed.

7.3 Areas for Further Work

Although the final system produced for this project was successful and could be used by small-medium hotel owners. I still believe there are great improvements which could be done. This section of the report will outline some improvements/recommendation for further work.

7.3.1 Social Element

In today's society, before people are willing to purchase anything, we find that they like to look at reviews about a particular product, and afterwards they like to rate the particular product. Further work which could be done to improve this system, would be to add this social element to the system. One way to do just this is by adding a rating and commenting system to every hotel room and to the hotel in general. This will allow viewers who are thinking about reserving a room at the hotel to be more inclined too if they know others have been there and given it a positive rating.

7.3.2 Online Payments

In terms of the greatest functionality which could be implemented to improve the system, would have to be the ability to pay online. As it currently stands the system has no method of payment, there is an assumption that the payment should be made upon checking out of the hotel, when the

customers are at the hotel. I.e. it is not done online. However, the majority of people who use such services like the added value of convenience, therefore adding online payment as an area for further work which could be done to improve the system, Is a functionality which I think would be most beneficial.

7.3.3 Other Functionality

There are almost endless possibilities of adding functionality which could be listed here. But, one of most importance has to be a greater search function. Again it comes down to people who use this type of service would like to have the ability to search for a room using multiple criteria and not just the check in and check out date , as is the method currently.

7.3.4 Mobile Version

Finally, an aspect not seen on the hotel market today, a mobile version of the system could be ported to smartphones, which would allow users to reserve a hotel room on the go. Additionally, allowing managers of the hotel to manage certain aspects of the system on the go. There is a lot of speculation and evidence to provide that the mobile market is increasing rapidly, therefore to ignore this recommendation for further work, would only be seen as a hindrance in the future, as to many the mobile market is seen as serious business.

7.4 Personal Gains / Challenges

In my opinion I believe that this project of developing a web-based information system accompanied with a report was successful. From the first stage to the last stage, I believe I have learnt a lot from undertaking a project such as this.

In conducting a literature review from this project, it has taught me how important research can actually be in producing something of great quality. Not only this, it has broadened my knowledge of every aspect of the development lifecycle from analysis to testing. I have learnt many methodologies and techniques which I had only heard of before. But, with this project I was able to carry out some of these techniques and critically assess them in my own work. Critically evaluating other systems was a great challenge to me, as it is very hard to know the evaluation criteria for every system, as each system although they seem very similar they can be very far apart. The amount of information

available regarding this topic is vast. Therefore, I believe there could have been a lot more research to undertake. However, it is very difficult in knowing what information should be included and what shouldn't be.

The analysis and design stage was a crucial point to my project. At first it was very hard to determine which methodologies and technique belong to the analysis and which to the design. As from a lot of designs techniques, analytical information can be produced from them, to assess an organisations system. Furthermore, this stage of the project taught me the importance of requirements gathering and producing concrete specifications, as these specifications will be securitized throughout the rest of the project. From the literature review conducted, it helped me find technique regarding this issue of producing requirements. Thus, I learnt about different methods such as agile development and others, to ease this process. Additionally knowledge I gained from university became a great asset to this project, such as the knowledge I had of UML. I was able to produce designs of the system which would later help me develop it, in such a way that it was efficient and reliable.

Then came the development stage, this stage took the longest out of the entire project as expected. The development stage was also the most challenging, as I had to learn many new things which I had never stepped foot into before. Web-design and web-development knowledge such as, HTML and CSS. Although having said that, I did have minimal prior knowledge of HTML before. However, I still had to learn ASP.NET, a scripting language from scratch, this was the core language throughout the entire project, which made it possible for the functions to work the way they did. Finally, knowledge I learnt about C# become useful in the development stage. However, I still had to learn how to integrate all of these languages together. What I found easy during the development stage was the development of the database, I really do feel that the knowledge gained from university about SQL albeit SQL used in ORACLE and not in Microsoft SQL server, which is the programed used for this project. Nevertheless it was still vital and as I had learnt so much regarding this aspect, it was by far the easiest to implement, but not without some prior research of course. However, there were many problems encountered during the development stage, where my knowledge on how to research solutions became vital to the progress of the project.

Testing the final stage of the project, I had very minimal prior knowledge of how to test a system before the literature. However, after the literature review I still managed to learn about 2-3 core techniques on how to test a system, specifically a web-based information system. In conducting the testing phase it really showed me how much you can learn about your system by testing it. The

amount of problems and errors I had found due to whatever reason had only scratched the surface of what was provided in this report. The problem with testing phase is when does testing become too much, although I never found this a problem. Testing the system made it more reliable, therefore it changed my thought process and showed me how crucial this was.

Evaluating the system was a very tricky part for me, as there are so many evaluation criteria you can test the system upon. Though this stage was quite interesting as I got to see what others thought about my system. Not only this, the participants who took part really honed in on the ideas I already had on how to make this system better.

Overall the project helped me gain a lot of valuable knowledge I will take with me. If I had to start this project again from scratch, I would probably use more of a mix of the development methodologies. I believe with this type of project, developing/producing something prior can really oversee what needs to be built in the rest of the system. I would also try to concentrate more on making the system more aesthetically pleasing. In terms of functionality it would have been nice to add in more social aspect to it such as rating system etc., all elements discussed in the "areas for further work" come to mind.

This project has been a great learning experience for me. I have vastly improved my technical abilities (HTML, CSS, ASP.NET, C#, SQL, SQL Server, JQuery and Ajax). Additionally my general academic skills have improved, skills such as academic writing, conducting research and evaluating have improved significantly. If I were asked to produce something like this again, I would feel a lot more confident working on a project similar to this.

7.5 Project summary

This project has explored the solution to developing web-based information, with regards to using common practising in analysing, designing and development the system. The goal was to provide a reliable and efficient system to managing a hotel.

The system was built by following these common techniques, the system was then evaluated against the objectives set at the beginning of the project, it was found that the system successfully met those objectives. However, there are still areas for improvements and further work which could be conducted.

References

AgileManifesto, 2001. *AgileManifesto*. [Online] Available at: agilemanifesto.org [Accessed 20 December 2012].

American Hotel Lodging Association, n.d. *Information Center*. [Online] Available at: http://www.ahla.com/content.aspx?id=4072 [Accessed 11 November 2012].

Avison, D & Fitzgerald,G, 2003. *Information Systems Development: Methodologies, Techniques and Tools*. 3rd ed.

BEVAN, NIGEL, 2006. *International Standards for HCI*. [Document] London, United Kingdom: NIGEL BEVAN [Accessed 26 December 2012].

Beynon Davies, 2009. [Online] Available at: http://www.palgrave.com/business/beynon-daviesbis/docs/chapter1.pdf [Accessed 11 November 2012].

Dawson, C, 2009. *Projects in Coomputing and Information Systems: A Student's Guide*. 2nd ed.

Debra Paul and James Cadle, 2010. Modelling Requirements. In D. Paul, D. Yeates & J. Cadle, eds. *Business Analyst*. Second Edition ed. p.186.

Dix, A; Finlay, J; D.Abowd,G; Beale,R, 2004. Design Process. In *Human-Computer Interaction*. 3rd ed.

Doran, G. T., 1981. There's a S.M.A.R.T. way to write management's goals and objectives. In Doran, G.T. *Management Review, Volume 70, Issue 11*. pp.35-36.

Groff,J R & Weinberb, P N, 2002. In *SQL The Complete Reference*. 2nd ed.

HNComputing, 2007. *sqa.org.uk*. [Online] Available at: http://www.sqa.org.uk/e-learning/SDM01CD/page_02.htm [Accessed 12 December 2012].

Kenneth Hess, 2010. *Top 10 Enterprise Database Systems to Consider*. [Online] Available at: http://www.serverwatch.com/trends/article.php/3883441/Top-10-Enterprise-Database-Systems-to-Consider.htm [Accessed 5 January 2013].

Krug, S, 2006. *Don't make me think. A common sense approach to web usability.*

Malcolm Eva, 2010. Requirements Gathering. In D. Paul, D. Yeates & J. Cadle, eds. *Business Analysis*. Second Edition ed. p.149.

McConnell, 1996. *Rapid Development: Taming Wild Software Schedules*.

Microsoft Corporation, 2005. *Black Box and White Box Testing for Application Blocks*. [Online] Microsoft Corporation: Microsoft Corporation Available at: http://msdn.microsoft.com/en-us/library/ff649503.aspx [Accessed 02 February 2013].

Nielsen, J & Tahir, M., 2001. *Homepage Usability*.

Peter Chen, 1976. The Entity Relationship Model: Toward a Unified View of Data. In *ACM Transactions on Database Systems*., 1976.

Scott W. Ambler, 2005. *agilemodeling*. [Online] Available at: http://www.agilemodeling.com/ [Accessed 01 December 2012].

SEI Report, Glossary, n.d. *sei*. [Online] Available at: http://web.archive.org/web/20070903115947/http://www.sei.cmu.edu/publications/documents/03.reports/03tr002/03tr002glossary.html [Accessed 12 November 2012].

Siau, K & Lee,L, 2004. Are usecase and class diagrams complementary in requirements analysis?. In *Requirements Engineering*. pp.229-37.

w3schools, N/A. *Server-side Scripting*. [Online] Available at: http://www.w3schools.com/web/web_scripting.asp [Accessed 10 January 2013].

Appendices

Appendix 1 – Terms of Reference

- **Title**

The title of the project I am proposing is “Hotel Management System: Analysis, Design and Development”.

- **Project Background/Purpose**

In this technology age, there are many people who can produce a system. However, producing a reliable, durable, maintainable and manageable system is a different story. Although many systems have many different concepts, the process of producing a hotel system in comparison to any other system is not all that different. Having said that, there are indeed different processes of producing such systems, which will be discussed later. Taking this into consideration, anyone looking to analyse, design and develop a system, should follow some common methodologies to achieve the above.

Producing a hotel system in particular can be challenging, there will be many aspects to take into consideration, not only on the development stage, but also on the analysis and design stages. Understanding what techniques to use for each stage is vital to producing a successful system.

Research:

Stage1:

1. To investigate current Hotel Systems. (Information Required, Development Methodologies, Implementation Tools/Techniques, And Common Functionality).
2. To understand and learn implementation techniques for developing a Hotel System. (Databases, Web Design and Web Development).
3. To research Systems Analysis and Design Techniques (Requirements Gathering, Specification Design, Systems Modelling and Systems Design).

Systems Analysis and Design:

Stage1:

1. Produce documents for requirements gathering process (Questionnaires and Interview Questions).
2. Gather Requirements from “client”.
3. Analysis of different stakeholder requirements.
4. Use case specification and diagram design of requirements of the system.
5. To produce specification of functional and non-functional requirements of the system.

Stage2:

6. To produce a Data Flow Model of the expected system.
7. To produce Business Process Modelling (Swim Lanes) of expected system.
8. Outline data fields required, carry out relational database analysis and finally produce data dictionary.
9. Produce a class model of expected system.

10. Produce Entity Relationship Model of the expected system.
11. Develop State Model of the system.
12. Business Systems Analysis of current and expected system(s).
13. Design wireframe models of expected system including all functionality.

Development/Implementation:

Stage1

1. Develop front end web-site based on design stage of wireframe model
2. Develop backend database.

Stage2

3. Implement a user registration, login and profile system.
4. Implement hotel information (Room details etc.).
5. Implement a room search function.
6. Implement control panel system to allow authorised users to manage hotel with use of add, edit and delete functions.
7. Implement booking system to allow users to reserve a room.

Testing:

Stage1

1. Produce test strategies.
2. To Identify Test Cases
3. To carry out Tests of Test cases

Evaluation:

Stage1

1. Evaluate project against initial aims and objectives.
2. Evaluate project against stakeholder requirements
3. Evaluate project against personal performance.

- **Requirements**

The requirements of this project are for someone to be able to create a system based upon a client's requirement. To have an understanding of what it takes to produce a system and finally be technically minded enough to develop this system or to be willing to learn how to create such a system; however this is based upon what the client actually requires. In terms of resource requirements (hardware/software) the resources to produce a system can be found at the university or downloaded free online therefore is no further resource requirement costs are needed.

- **Potential Risk**

The potential risk of producing this project is the client may not like the outcome of the system therefore the need to create detailed requirements and specification of the system is crucial to ensure the system being created is designed and developed with the need of the client as the main focus. Although I am using a new language and technology to produce this project which I have little experience in, as a fall back to the project it may be implemented as a global application where each department of the hotel has access to this application allowing information to be directly inputted.

- **Course-Specific Learning Outcomes**

1. *To enable students to gain an advanced understanding of database management systems, and to develop advanced skills in their exploitation.*

The project I have proposed will enable me to learn new things and enhance on the database development skills I have learnt. With the creation of a Hotel System I will need a database to store various data thus the learning outcome of this is to develop my skills in SQL. Also I will need to manage the database therefore I will need to learn a database management application the one I will be learning is Microsoft SQL Server 2008/2010. To exploit and manipulate this database I will need to enhance my skills of T-SQL I will also be learning how to use SSRS (SQL Server Reporting Services).

2. *To develop the ability to design, specify, construct and maintain software.*

With the development of a Hotel System I will need to be able to design a website for the client. Therefore I will be developing my skills on producing UML diagrams and wireframes of the system as well as producing specifications. The development of the website will require of me to enhance my skills in html and learn how to use CSS to design the website. The construction of the hotel system will require me to develop my skills in C# as well as the .Net Language.

3. *To equip students with skills and attributes that will enhance employment opportunities.*

With this project I will learn new skills and enhance my current skills which will make me more employable. The skills I will learn and enhance for the purpose of this project includes; SQL/T-SQL for the development and manipulation of the hotel database, SQL Server 2010 for the management of the hotel database, HTML for the creation of the website as well as CSS for the design of the website, ASP.Net for the scripting of the website, ADO.Net for the integration/manipulation of the database and C# for the development. Finally I will be enhancing my skills in systems analysis by producing various different analysis/development techniques of my system which include UML techniques, business process modelling techniques and other business systems analysis techniques such as creating specification and requirements gathering.

- **Timetable and Deliverables**

At the end of this project I intend to submit a system for a hotel which meets the requirements of the client and a report which documents the analysis, design and development of the system.

- Hotel Systems Analyst Stage1 = Gather Requirements from client.
- Hotel Systems Analyst Stage2 = Produce Specification for the system.
- Hotel Systems Analyst Stage3 = Researching possible solutions for building the system.
- Hotel Systems Analyst Stage4 = Modelling the system using Unified Modelling Language and Business Process Modelling.

- Hotel Systems Design Stage1 = Produce wireframe modelling of expected system.
- Hotel Systems Design Stage2 = Begin HTML programming of the website.
- Hotel Systems Design Stage3 = Designing the look of the HTML website using CSS.

- Hotel Systems Development Stage1 = Development of the database using Microsoft SQL Server.
- Hotel Systems Development Stage2 = Integration of ASP.Net into the system.
- Hotel Systems Development Stage3 = Integration of the database using ADO.net and C# Programming.
- Hotel Systems Development Stage4 = Finalising the development of the System

Project TimeTable

			28-Sep-12	05-Oct-12	12-Oct-12	19-Oct-12	26-Oct-12	02-Nov-12	09-Nov-12	16-Nov-12	23-Nov-12	30-Nov-12	07-Dec-12	14-Dec-12	11-Jan-13	18-Jan-13	25-Jan-13	01-Feb-13	08-Feb-13	15-Feb-13	22-Feb-13	22-Feb-13	08-Mar-13	15-Mar-13	22-Mar-13	19-Apr-13
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
START	24-Sep-12	1	-																							
Research	01-Oct-12	2		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X				
Terms of Reference	08-Oct-12	3			X																					
Hotel System Anaylsis S1	15-Oct-12	4				-	X																			
Hotel System Anaylsis S2	22-Oct-12	5					-	X																		
Hotel System Anaylsis S3	15-Oct-12	6						-	X																	
Threshold Meeting 1	05-Nov-12	7							X																	
Literature Review	12-Nov-12	8								X																
Hotel System Anaylsis S4	19-Nov-12	9									-	-	-	-	-	-	-	-	-	-	-	-	X			
Interim Report	26-Nov-12	10										X														
Hotel System Design S1	03-Dec-12	11											-	X												
Hotel System Design S2	10-Dec-12	12												-	-	-										
Hotel System Design S2 + S3	07-Jan-13	13													-	-	-	-	-	-	-	-	-	X		
Hotel System Development S1	14-Jan-13	14														-	-	-								
Hotel System Development S2	21-Jan-13	15															-	-	-							
Hotel System Development S3	28-Jan-13	16																-	-	-						
Hotel System Development S4	04-Feb-13	17																	-	-	-					
Report Structure	11-Feb-13	18																		X						
Threshold Meeting 2	18-Feb-13	19																			X					
Hotel System Development S4	25-Feb-13	20																				-	-	-	X	
Testing	04-Mar-13	21																					-	X		
Finalising	11-Mar-13	22																						-		
Project Report	18-Mar-13	23																							X	
Submission	15-Apr-13	24																								X

Key	
"-/X"	"Start/Finish"
	"Milestones"
	"Tasks"

Appendix 2 – Ethics Form

Manchester Metropolitan University

ETHICS CHECK FORM

This checklist must be completed for every project. It is used to identify whether there are any ethical issues associated with your project and if a full application for ethics approval is required. If a full application is required, you will need to complete the 'Application for Ethical Approval' form and submit it to the relevant Faculty Academic Ethics Committee, or, if your research falls within the NHS, you will need to obtain the required application form from the National Research Ethics Service available at www.nres.npsa.nhs.uk/ and submit it to a local NHS REC.

Before completing this form, please refer to the University's Academic Ethical Framework (www.rdu.mmu.ac.uk/ethics/mmuframework) and the University's Guidelines on Good Research Practice (www.rdu.mmu.ac.uk/rdegrees/goodpractice.doc).

Project and Applicant Details

Name of applicant (Principal Investigator):	Kamran Khan
Telephone Number:	
Email address:	10167313@stu.mmu.ac.uk
Status: **(please circle as appropriate)**	Undergraduate Student Postgraduate Student (Taught or Research) Staff
Department/School/Other Unit:	School of Computing, Mathematics and Digital Technology
Programme of study (if applicable):	Bsc(hons)Computing
Name of supervisor (if applicable):	George Fakas
Project Title:	Hotel System: Analysis Design and Development
Does the project require NHS Trust approval? If yes, has approval been granted by the Trust? Attach copy of letter of approval.	NO

Ethics Checklist (Please answer each question by ticking the appropriate box)

	Yes	No	N/A
1. Will the study involve recruitment of patients or staff through the NHS, or involve NHS resources? If yes, you may need full ethical approval from the NHS.		X	
2. Does the study involve participants who are particularly vulnerable or unable to give informed consent (e.g. children, people with learning disabilities, your own students)?		X	
3. Will the study require the co-operation of a gatekeeper for initial access to the groups or individuals to be recruited (e.g. students at school, members of self-help group, nursing home residents)?		X	
4. Will the study involve the use of participants' images or sensitive data (e.g. participants personal details stored electronically, image capture techniques)?		X	
5. Will the study involve discussion of sensitive topics (e.g. sexual activity, drug use)?		X	
6. Could the study induce psychological stress or anxiety or cause harm or negative consequences beyond the risks encountered in normal life?		X	
7. Will blood or tissue samples be obtained from participants?		X	
8. Are drugs, placebos or other substances (e.g. food substances, vitamins) to be administered to the study participants or will the study involve invasive, intrusive or potentially harmful procedures of any kind?		X	
9. Is pain or more than mild discomfort likely to result from the study?		X	

10. Will the study involve prolonged or repetitive testing?		X	
	Yes	**No**	**N/A**
11. Will it be necessary for participants to take part in the study without their knowledge and informed consent at the time (e.g. covert observation of people in non-public places)?		X	
12. Will financial inducements (other than reasonable expenses and compensation for time) be offered to participants?		X	
13. Is there any possible risk to the researcher (e.g. working alone with participants, interviewing in secluded or dangerous)?		X	
14. Has appropriate assessment of risk been undertaken in relation to this project?	X	X	
15. Does any relationship exist between the researcher(s) and the participant(s), other than that required by the activities associated with the project (e.g., fellow students, staff, etc)?		X	
16. Faculty specific question, e.g., will the study sample group exceed the minimum effective size?		X	

If you have ticked 'no' or 'n/a' to all questions, attach the completed and signed form to your project approval form, or equivalent. Undergraduate and taught higher degree students should retain a copy of the form and submit it with their research report or dissertation (bound in at the end). MPhil/PhD, and other higher degree by research, students should submit a copy to the Faculty Research Degrees Sub-Committee with their application for registration (RD1) and forward a copy to their Faculty Academic Ethics Committee. Members of staff should send a copy to their Faculty Academic Ethics Committee before commencement of the project.

If you have ticked 'yes' to ***any*** of the questions, please describe the ethical issues raised on a separate page. You will need to submit your plans for addressing the ethical issues raised by your proposal using the 'Application for Ethical Approval' form which should be submitted to the relevant Faculty Academic Ethics Committee. This can be obtained from the University website (http://www.rdu.mmu.ac.uk/ethics/index.php).

If you answered 'yes' to question 1, you may also need to submit an application to the appropriate external health authority ethics committee, via the National Research Ethics Service (NRES), found at http://www.nres.npsa.nhs.uk/, and send a copy to the Faculty Academic Ethics Committee for their records.

Please note that it is your responsibility to follow the University's Guidelines on Good Research Practice and any relevant academic or professional guidelines in the conduct of your study. **This includes providing appropriate information sheets and consent forms, and ensuring confidentiality in the storage and use of data.** Any significant change in the question, design or conduct over the course of the research should be notified to the relevant committee (either Faculty Academic Ethics Committee of Local Research Ethics Committee if an NHS-related project) and may require a new application for ethics approval.

Approval for the above named proposal is granted

I confirm that there are no ethical issues requiring further consideration. *(Any subsequent changes to the nature of the project will require a review of the ethical consideration(s).)*
Signature of Supervisor (for students), or Manager (for staff): _____
Date: _____

Approval for the above named proposal is not granted

I confirm that there are ethical issues requiring further consideration and will refer the project proposal to the Faculty Academic Ethics Committee.
Signature of Supervisor (for students), or Manager (for staff): _____
Date: _____

Ethics Matters

Separate page for ethical issues:-

Ethics Matters

Appendix 3 – Interim Report

According to my terms of reference, in the aims and objectives I described, I was expected to have completed and started on certain aspects of my project, as outlined in the Gantt chart. Due to the severe depth of my project in the written part and also the implementation, I have had to make some changes in terms of rearranging slightly when I will undertake each task. Not only this, but I have also redefined clear and in depth the aims and objectives of my project. Due to this I will have to amend my Gantt chart.

- **Re-defined Aims and Objectives**

The reason I have redefined the aims and objectives is that I feel the previous objectives were not clear enough targets to aim for over a certain period. Therefore by being broad it is difficult to see the progress of the project.

Aims & Objectives: Aim is to develop an Online Information-System specifically a Hotel System, which meets the requirements of the client. As well as documenting in depth, the entire systems development lifecycle for this system. The objectives have been redefined to reflect SMART (Specific, Measurable, Realistic and Timely) targets, with the aid of the development lifecycle.

Research

1. Research Analysis, Design, Development, Testing and Evaluation Techniques.
2. Research current and previous Hotel Systems.

Analysis & Design

1. Gather and identify requirements.
2. Produce specification for functional and non-functional requirements.
3. Produce system Use case specification and diagrams from gathered requirements.
4. Produce system Data Flow model with explanation and justification.
5. Produce system Business Process/State Model with explanation and justification.
6. Produce system Data Dictionary with explanation and justification.
7. Produce system Relational Database Analysis with explanation and justification.
8. Produce system Class Model with explanation and justification.
9. Produce system Entity Relationship Model with explanation and justification.
10. Produce Wireframe designs of expected system with explanation and justification

Implementation

1. Develop front end to system.
2. Develop back end to system.
3. The system should allow users to login/register
4. The System should allow users to search for a room and reserve one.
5. The System should allow users to access their profile page.
6. The System should allow users to cancel reservations.
7. The System should allow users to update any details, including password.
8. The System should allow for an admin section
9. The System should allow admins to manage Rooms.
10. The system should allow admins to manage Reservations
11. The System should allow admins to manage Users

Testing

1. Create test strategies.
2. Identify test cases.
3. Carry out system testing.
4. Test system in terms of validation and compatibility

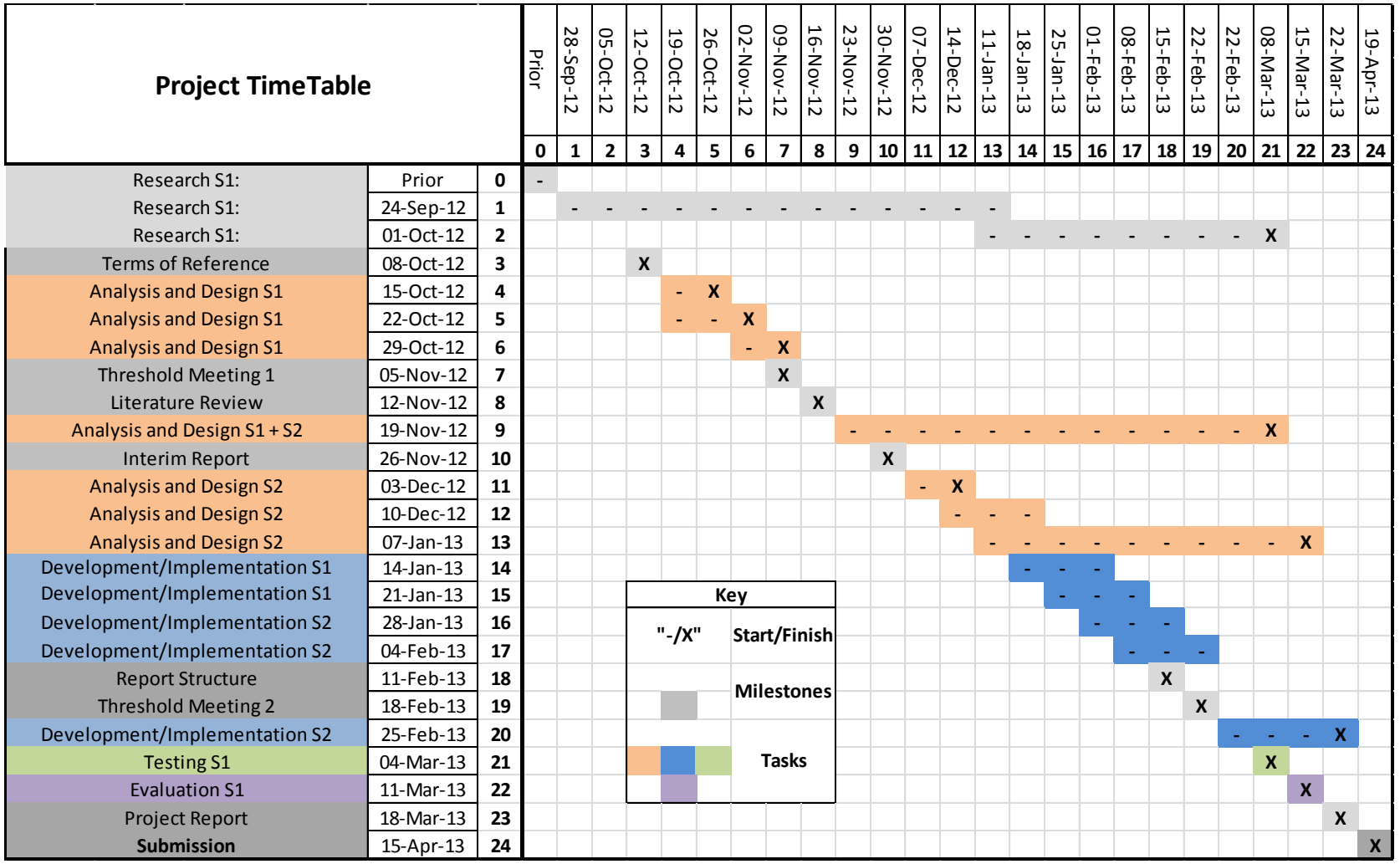
Evaluation

1. Produce Evaluation criteria.
2. Evaluate project against initial aims and objectives.
3. Evaluate project against initial requirements.
4. Evaluate project against user's acceptance.

Conclusion

1. Evaluate personal performance and improvements.

- **Amended Time Table**



- **Current Progress**

In this section of the interim report I will outline my current progress, detailing what I have done so far.

Terms of Reference: Thus far I have completed my term of reference. The terms of reference details the title of my project, the background information to my project, describing clearly the aims and objectives of the project, assessing the requirements needed to undertake such project, detailing any potential risks I may encounter with the project and finally producing deliverables through a timetable Gantt chart.

Literature Survey: I have also conducted a literature survey. The literature survey consists of an introduction outlining the research I am undertaking, as well as proposing my potential question. It then consists of a review of Systems Analysis and Design, outlining the importance of systems analysis design and describing various techniques. I then do a review of existing hotel systems, which includes an analysis of different methodologies used for development as well as a comparison and contrast of different types of hotel systems. Furthermore I review some system implementation technologies/tools to developing a hotel system. Such as reviewing different software's I could use, different databases I could implement and various other ways of developing such system. And Finally I have concluded the literature review with my approach.

Research: Alongside my literature survey I have done some prior research. The research consisted of me learning some basics to developing a system of this nature. I had prior knowledge of relational database's however to fully undertake this project, I've had to learn some of the basics of web development and web design.

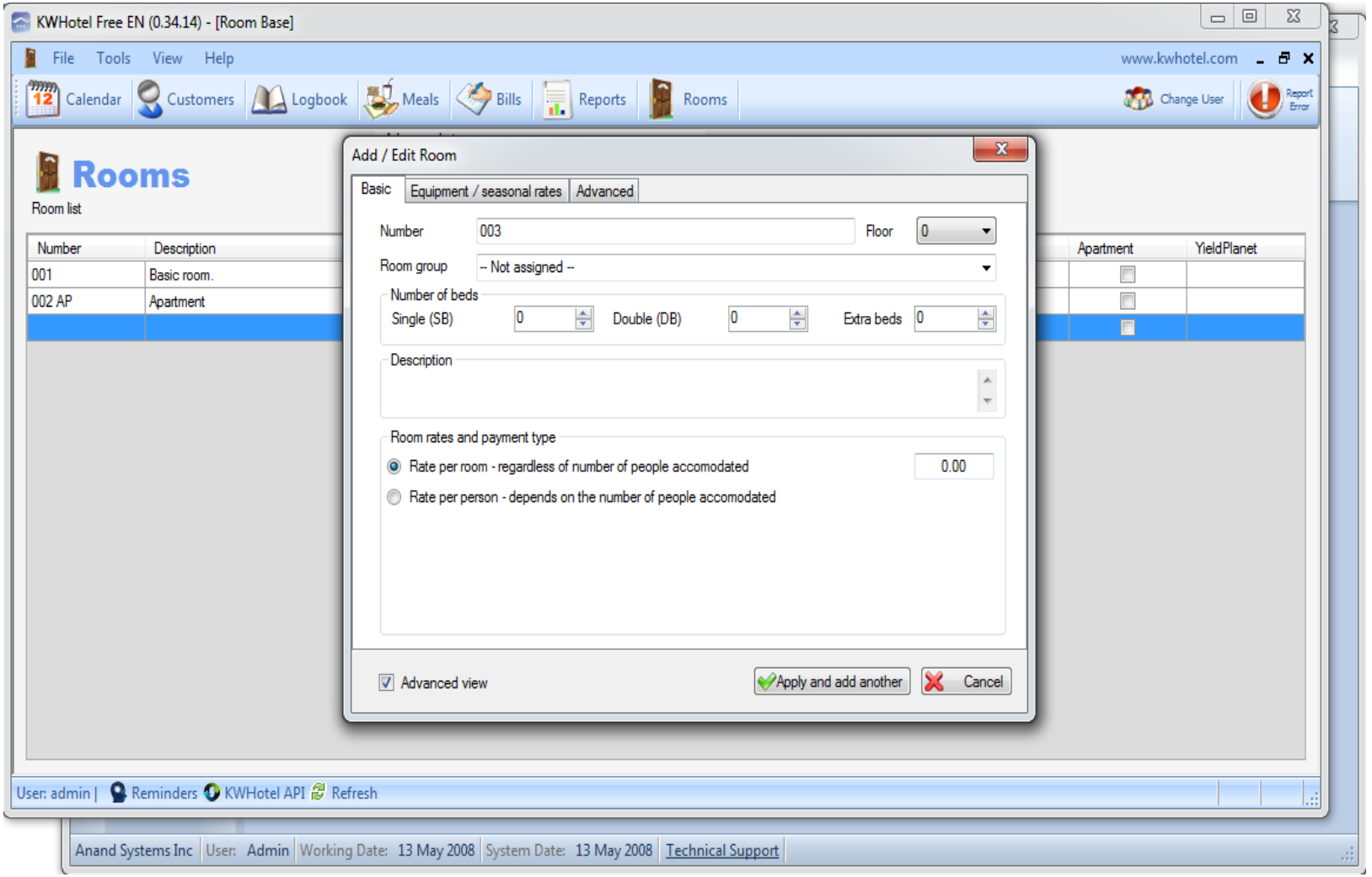
Systems Analysis and Design: I have started producing a systems analysis and design for the proposed system. Currently I have produced requirement gathering tools such as producing a questionnaire and some key interview questions from this I have gained the requirements from the client and different stakeholders. I have then catalogued each requirement in functional and non-functional groups. Then producing use case specification and model from these requirements. Therefore, I have nearly completed stage 1 of my systems analysis and design phase. I have also started stage 2 of the systems analysis and design stage. I produced a first draft of a data flow model

and business process model of the expected system as well as describing my data fields required for the system. As well as this I have also produced an initial entity relationship model for my project.

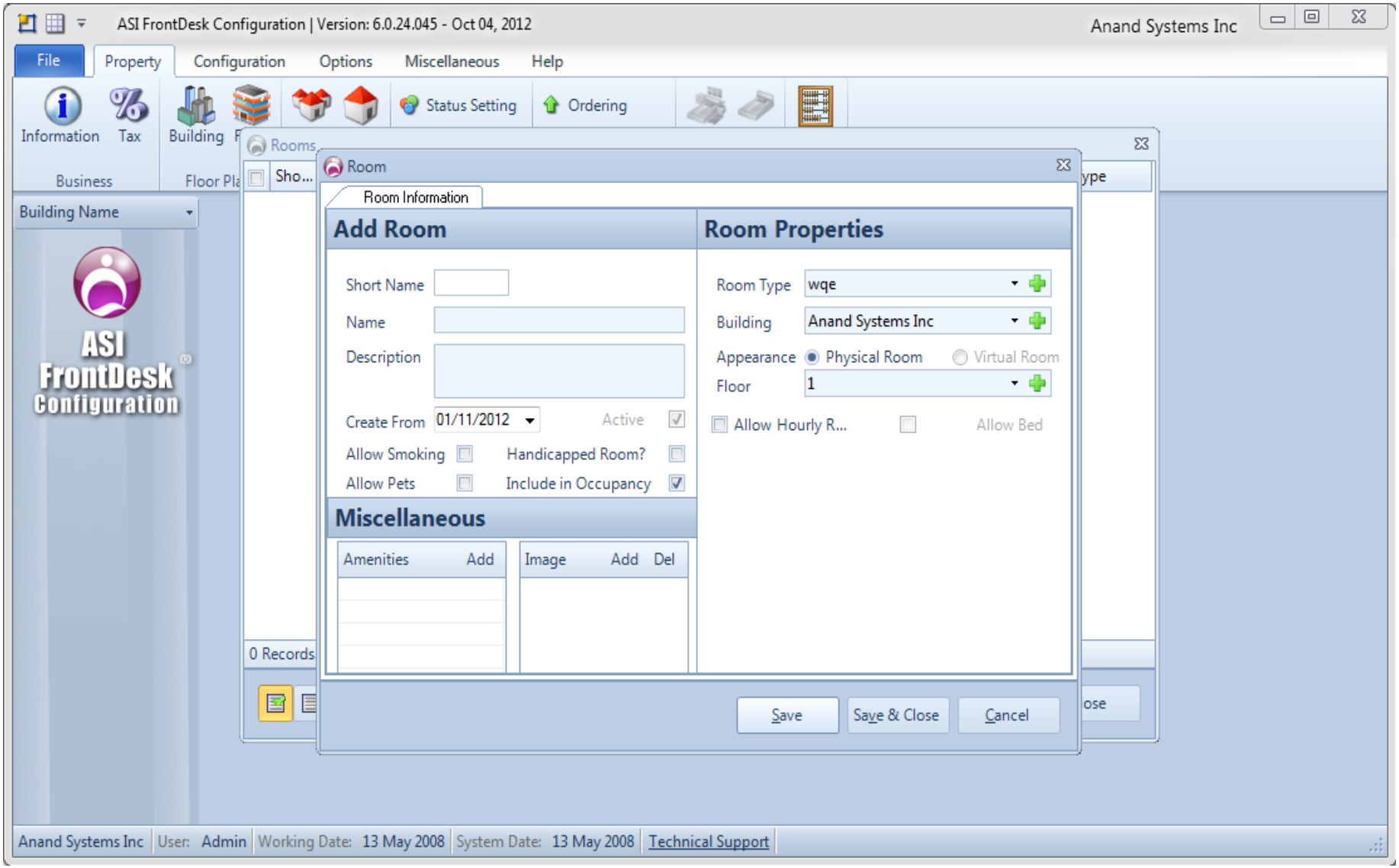
Implementation: Ahead of schedule I have also started to develop my web based information system. By implementing the layout of the website. The reason for this is I am incorporating a slight agile approach to implementation, by creating the layout of the website it will help me understand the capabilities and review the system. Encase of any potential changes or unforeseen circumstances which could occur when only implementing after all the analysis and design stage has finished.

Appendix 4 – Literature Review Hotels

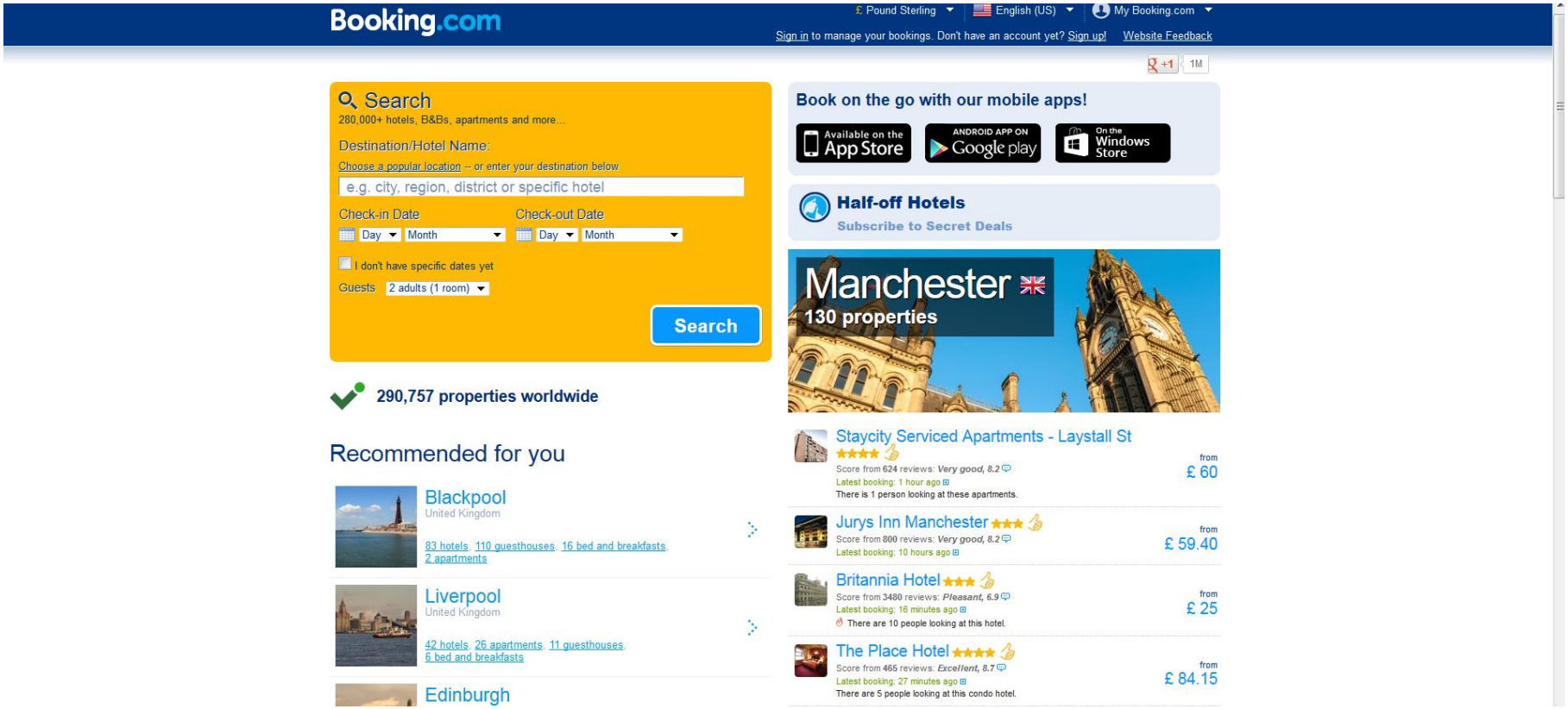
A4.1



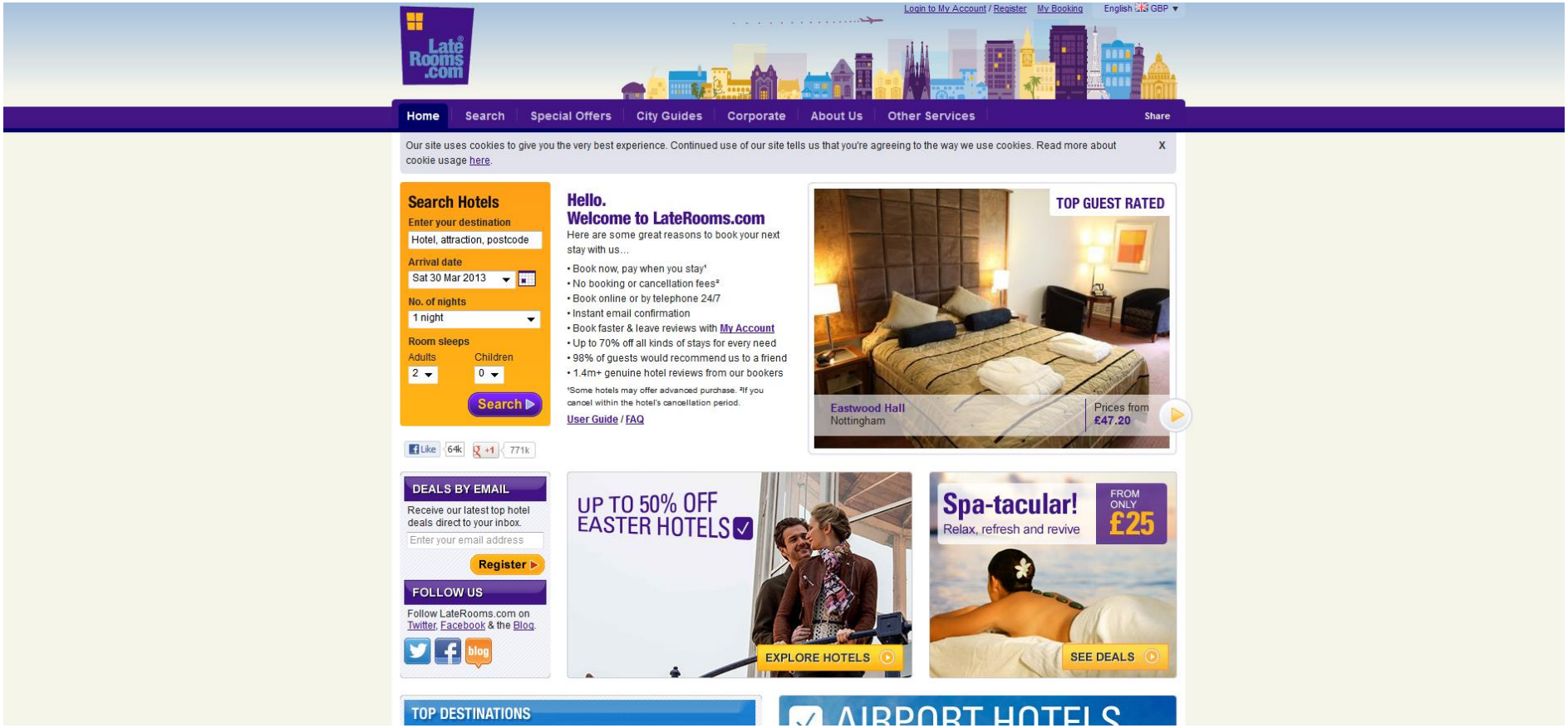
A4.2


ASI FrontDesk Configuration | Version: 6.0.24.045 - Oct 04, 2012
Anand Systems Inc
File
Property
Configuration
Options
Miscellaneous
Help
Information
Tax
Building
Status Setting
Ordering
Business
Floor Pla
Building Name
ASI FrontDesk Configuration
Rooms
Room
Room Information
Add Room
Short Name
Name
Description
Create From
01/11/2012
Active
Allow Smoking
Handicapped Room?
Allow Pets
Include in Occupancy
Miscellaneous
Amenities
Add
Image
Add
Del
Room Properties
Room Type
wqe
Building
Anand Systems Inc
Appearance
Physical Room
Virtual Room
Floor
1
Allow Hourly R...
Allow Bed
0 Records
Save
Save & Close
Cancel
Anand Systems Inc
User: Admin
Working Date: 13 May 2008
System Date: 13 May 2008
Technical Support

A4.3


Booking.com
£ Pound Sterling
English (US)
My Booking.com
Sign in to manage your bookings. Don't have an account yet? Sign up!
Website Feedback
+1 1M
Search
280,000+ hotels, B&Bs, apartments and more...
Destination/Hotel Name:
Choose a popular location -- or enter your destination below
e.g. city, region, district or specific hotel
Check-in Date
Check-out Date
Day Month
Day Month
I don't have specific dates yet
Guests 2 adults (1 room)
Search
290,757 properties worldwide
Recommended for you
Blackpool
United Kingdom
83 hotels, 110 guesthouses, 16 bed and breakfasts, 2 apartments
Liverpool
United Kingdom
42 hotels, 26 apartments, 11 guesthouses, 6 bed and breakfasts
Edinburgh
Book on the go with our mobile apps!
Available on the App Store
ANDROID APP ON Google play
On the Windows Store
Half-off Hotels
Subscribe to Secret Deals
Manchester
130 properties
Staycity Serviced Apartments - Laystall St
Score from 624 reviews: Very good, 8.2
Latest booking: 1 hour ago
There is 1 person looking at these apartments.
from £ 60
Jurys Inn Manchester
Score from 800 reviews: Very good, 8.2
Latest booking: 10 hours ago
from £ 59.40
Britannia Hotel
Score from 3480 reviews: Pleasant, 6.9
Latest booking: 16 minutes ago
There are 10 people looking at this hotel.
from £ 25
The Place Hotel
Score from 465 reviews: Excellent, 8.7
Latest booking: 27 minutes ago
There are 5 people looking at this condo hotel.
from £ 84.15

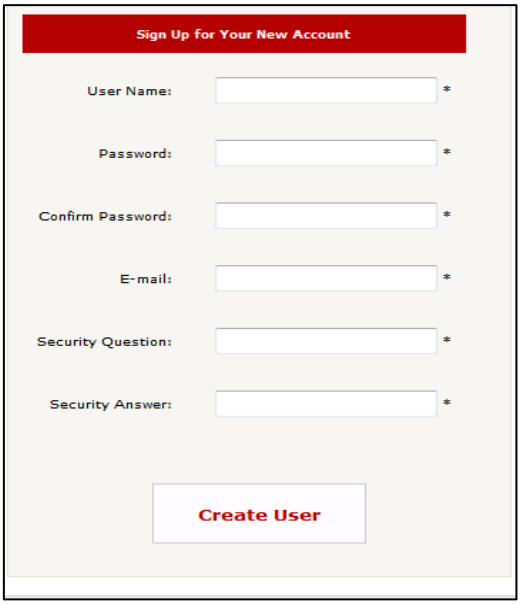
A4.4


Login to My Account / Register
My Booking
English GBP
Home
Search
Special Offers
City Guides
Corporate
About Us
Other Services
Share
Our site uses cookies to give you the very best experience. Continued use of our site tells us that you're agreeing to the way we use cookies. Read more about cookie usage here.
Search Hotels
Enter your destination
Hotel, attraction, postcode
Arrival date
Sat 30 Mar 2013
No. of nights
1 night
Room sleeps
Adults
Children
Search
Hello.
Welcome to LateRooms.com
Here are some great reasons to book your next stay with us…
• Book now, pay when you stay¹
• No booking or cancellation fees²
• Book online or by telephone 24/7
• Instant email confirmation
• Book faster & leave reviews with My Account
• Up to 70% off all kinds of stays for every need
• 98% of guests would recommend us to a friend
• 1.4m+ genuine hotel reviews from our bookers
¹Some hotels may offer advanced purchase. ²If you cancel within the hotel's cancellation period.
User Guide / FAQ
TOP GUEST RATED
Eastwood Hall
Nottingham
Prices from
£47.20
DEALS BY EMAIL
Receive our latest top hotel deals direct to your inbox.
Enter your email address
Register
FOLLOW US
Follow LateRooms.com on Twitter, Facebook & the Blog.
UP TO 50% OFF
EASTER HOTELS
EXPLORE HOTELS
Spa-tacular!
Relax, refresh and revive
FROM ONLY
£25
SEE DEALS
TOP DESTINATIONS
AIRPORT HOTELS

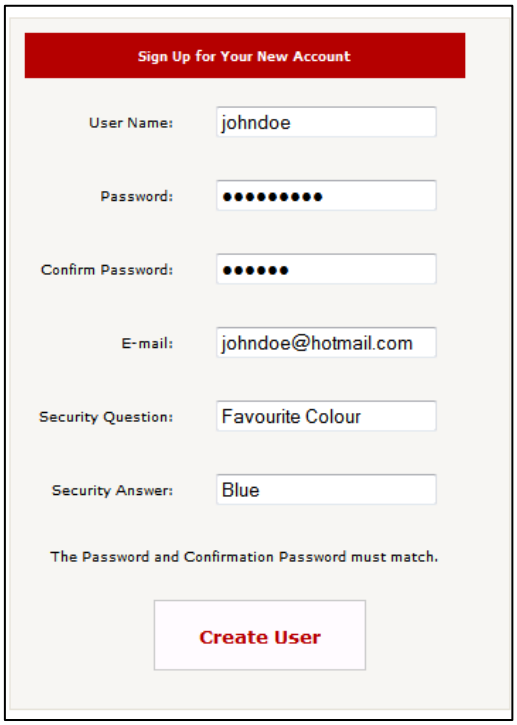
Appendix 5 – System Testing Evidence

Registration

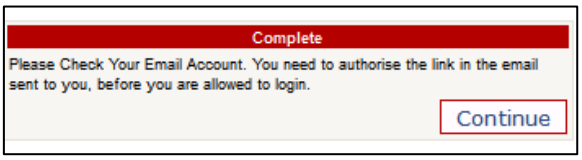
R1. – Details Left Blank

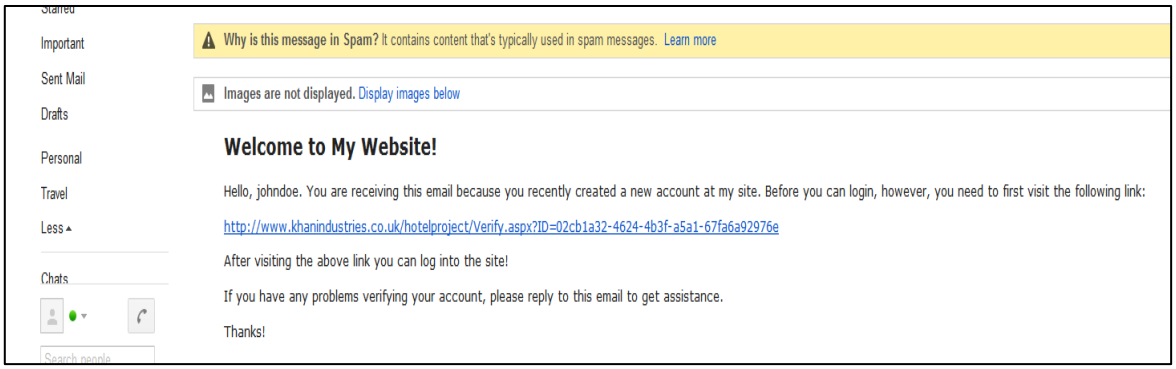


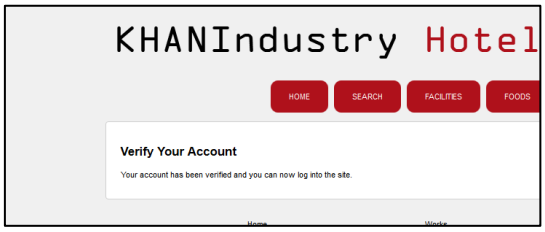
R2. – Wrong matching passwords



R3. – Successful Account Registration

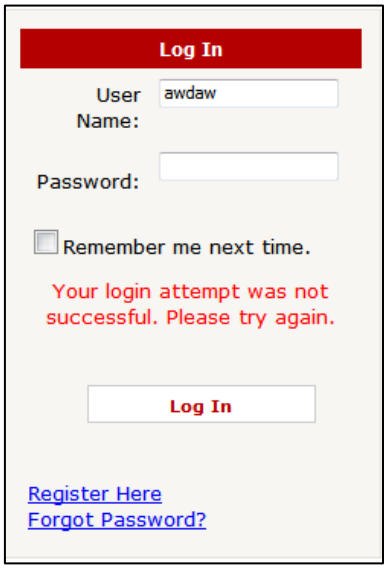






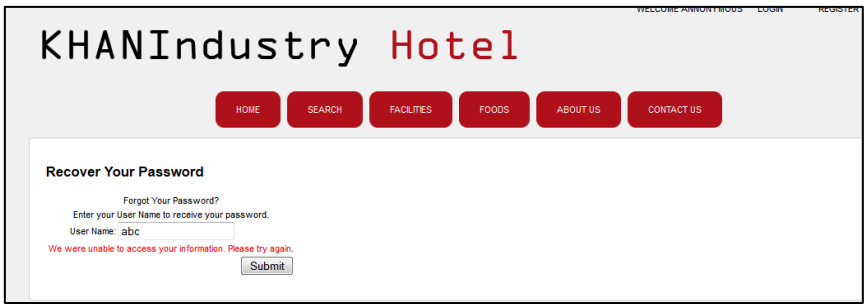
Login

L2. – Incorrect Password or Username

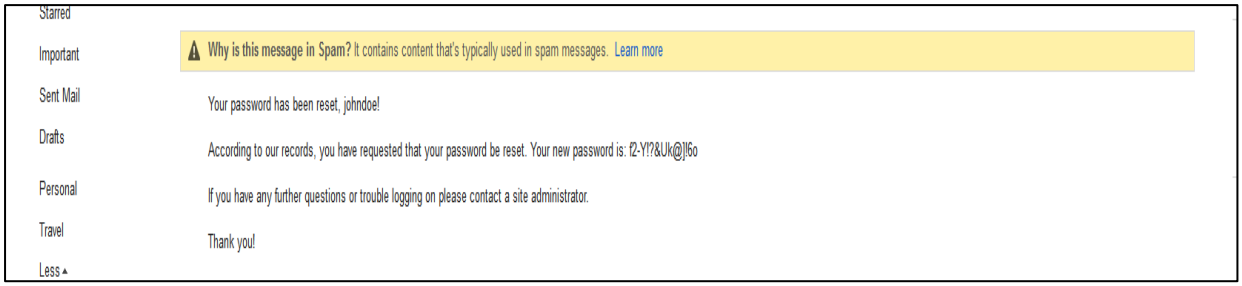


Forgot Password

F1. – Recover password for unknown username

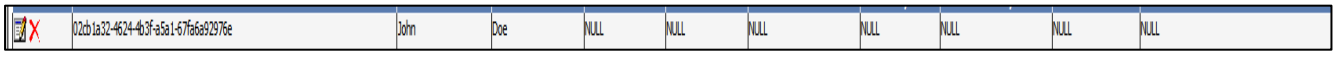


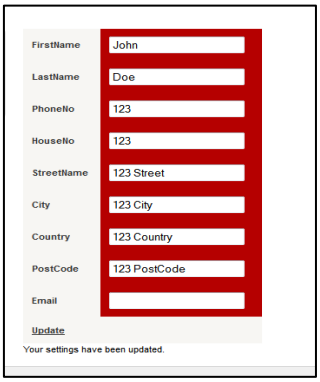
F2. – New Password Sent To Email

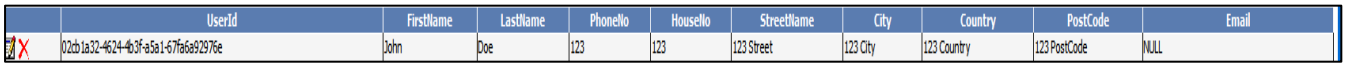


Update Profile

U1. – Update User Profile Details

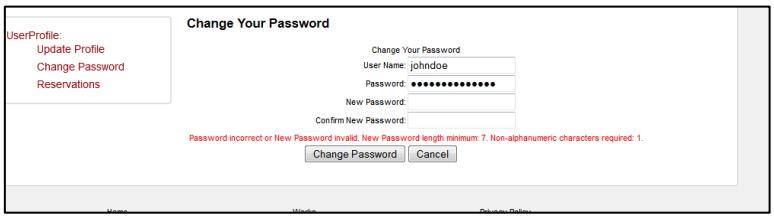






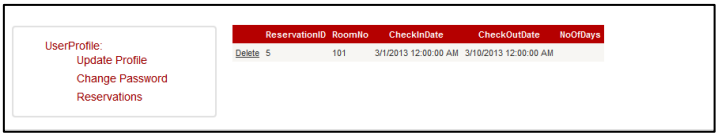
Change Password

C1. – Unknown existing password, or new password length invalid



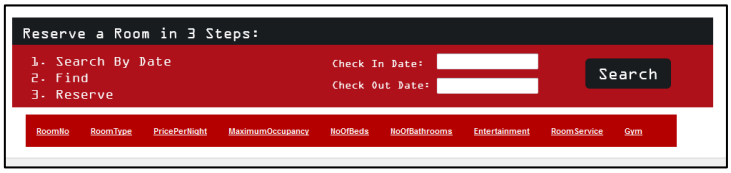
Cancel Reservation

CR1. – User tries to cancel/delete a reservation. Functionality Enabled.

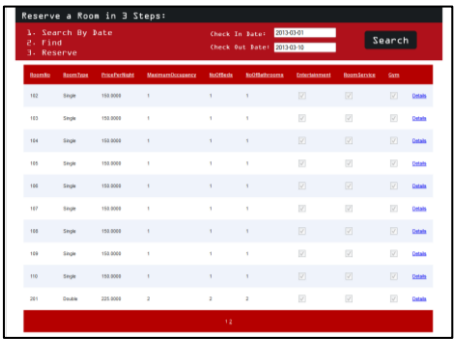


Search for Room

S1. – User tries to search for room with no dates or only 1 date.



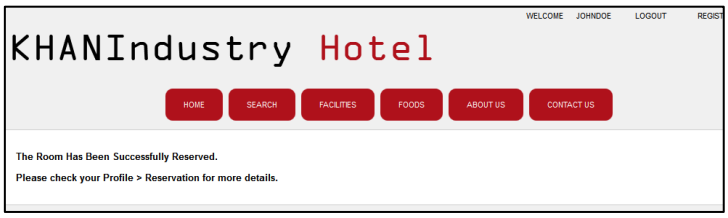
S3. – User tries to search for room with specified dates. Returns all rooms available.



Book Room For Reservation

B1. – Anonymous user tries to book a room for reservation. Error is displayed.

B2. – Registered Users tries to reserve a room.



This Page is Empty