Iteration0

Brandon Koepke, Alberto Saave, Garrick Vdl, Julia, Tyler Gibb, Justin Milanovic

Abstract

The following document describes the proposed Hotel Manager 4000 system including the intended goal of the project, a tentative release plan, the system requirements, critical success factors and the key development technologies to be used in the system's development.

Contents

1. Vision Statement

Hotel Manager 4000 has a single goal: developing the customer base. HM4000 helps Hotels maintain a mounting database of customers and their preferences, in addition to performing essential management task, such as room management, bookings and generating detailed reports quickly and easily with minimal training. The overall strategy is to create apparent value for your clients by studying their prior behavior in order to fashion a list of their future desires. Providing an added-value incentive, tied directly to sales, increases purchases among customers. Hospitality management is about growing the relationships with your guests, so spend more time pleasing guests than learning software.

2. Release Plan

The main features that we intend to implement for iteration 1 of our system is a user based form of access control. Customer users (guests at a hotel) for instance will have two basic forms of access control, anonymous customers and registered customers. Registered customers and anonymous customers will both have the ability to book and unbook rooms; on the other hand, registered customers will simply have the privilege of not having to enter their information every time they wish to book a room where as unregistered customers must do this every time they book a room, also registered customers will have the privilege of being able to unbook rooms where as unregistered customers will have to go through a sales associate. For users employed in the hotel company there are various levels of privilege, administers will have full access to the system as they represent the corporate scheme of the business so they essentially can do everything that all other users can do with the added benefit of adding hotels. Other users types Include Hotel managers, Sales Managers, Room Service Managers, Sales Representatives and Room Service Associates. Customer information access for associates is also intended to be implemented for iteration 1, this includes features such as adding,deleting and modifying customers as well as the ability to create a list of customers in order for an associate to look them up. Most of the above features are concerned with a partial established user control access mechanism that will be used in the future iterations.

The main features intended for iteration 2 of our system include the ability to allow appropriate users to book or unbook rooms, these users include both types of customer users, sales representatives, sales managers, and hotel managers. In order for this feature to work the user types will also have the ability to look for available booking time slots and choose the time slot most appropriate for them. Associates with the proper privileges working for the company will also have the ability of looking up or adding customer information, to select a customer in order to make a booking on that customers behalf. Hotel managers and sales managers will also have the ability of adding and deleting rooms from the hotel booking list in the case of a hotel restructuration. Also for iteration 2 hotel managers and sales managers should be

able to deal with adding,deleting and modifying associates who report to them in the employee hierarchy.

The main features intended to be implemented in iteration 3 of our system include the management of customer room service charges and anonymous user access. Customer users will be able to look at their current set of charges, this includes booking prices and rooms service charges. Sales associate, sales managers and hotel managers will also be able to look at a customer's billing status and will also be able to add charges regarding room bookings. Managers will also have the added privilege of removing charges. The room service associates and room service managers will have only the privilege to see charges to the customers room service bills, and the rooms service managers and possibly trusted associates will have the added benefit of removing room service charges from a customer's bill. If possible we may also implement a customer ticket system, where a customer can submit a ticket for complaints, complement and/or suggestions to which certain associates can respond to. Another possible feature which may be included in this iteration is for the ability of administrators to add additional hotels to the system.

3. Requirements

The requirements for the client and server for the hotel manager are different.

3.1. Client Requirements

Any operating system capable of running a browser from figure 1 will be supported. This includes Windows Vista and later, Mac OS X 10.5 and later and most Linux variants.

Browser	**Minimum Version**	**Layout Engine**
Firefox	18	Gecko
Google Chrome	24	WebKit
Safari	5	Webkit
Internet Explorer	9	Trident

Table 1: Supported Browsers

These web browsers were chosen because it enables us to support 93.34% of non-mobile web browsers as of October 2012 [2].

3.2. Server Requirements

The server is officially supported on Debian Squeeze. The system can be installed on other Linux and UNIX systems, but the installation will not be officially supported. Table 2 lists the necessary packages and their versions, some of which will have to be installed from backports.

Package	Minimum Version	Description	Rationale
Ruby	1.9.3	Programming Language	Ruby is a dependency of the Ruby on Rails framework.
RubyGems	1.8.24	Ruby Package Manager	RubyGems eases the installation of third-party ruby libraries. Using ruby gems, you can install the latest version of a library without having to use the system package manager.
Ruby on Rails	3.2	Full-Stack Web Application Framework	Most of the group is writing their paper assignment on Ruby on Rails and we currently only have one group member that has previous experience with Ruby on Rails. In order to write a well-reasoned paper on Ruby on Rails, the group decided that Ruby on Rails should also be used for the supplier assignment in order to gain experience with the framework. Ruby on Rails also favours convention over configuration which makes the creation of simple Create, read, update, and delete (CRUD) apps simpler.
Postgresql	8.4.13	Object-Relational SQL Database	Postgresql is the default database supported by Debian, and is also supported by Heroku which is a cloud platform as a service supporting Ruby. Many modern Linux distributions are also leaving MySql as their default database due to the recent Oracle acquisition of Sun (who previously owned MySql) [1].

Table 2: Required Software

4. Critical Success Factors

The hotel manager system must be easy to use, extendible, and scalable. These are non functional qualities that will aid users to have a better experience with the

system, and developers to be able to easily extend, improve, adapt, and add features to the system without breaking its main functionality. We will also like to make the webpage scalable so that the hotel can continue to serve their users better at each stage of their growing process as a company. By allowing the hotel management website to be able to grow together with the company, we are also allowing the hotel to take care of their old and new customers so that they can enjoy of all the amenities and services that the hotel offers in the present and in the future.

The hotel manager website must be easy to use for all the subjects that are going to use this service. This will initially include both staff and the hotel customers. Although it may be necessary to have some basic training to place orders, print records, and to perform other admin activities, we will want to make this as intuitive as possible. This website will certainly need to improve the effectiveness of the staff involved in performing these operations. Similarly, we want to make our customers to be able to find and book a room easy and fast. This will be achieved by focusing early on users and tasks. We will take into consideration and value the experience of the staff and customers with similar systems, and we will improve it upon that.

Besides making the website easy to use, we believe that in order to be successful we need to support extensibility of the system. This needs to be done because we are going to be adding many features on each iteration of the project, and we need to make sure that none of those features restrict the use of other possible features that we will want to implement at some point. One of the most important features that we will need to implement in the future is to allow an anonymous user to browse the contents of our webpage, so that they don?t have the necessity to provide us with sensitive information if they are just potential customers looking for alternatives. This should be done at some point because we acknowledge that some people don?t like the idea of registering for every single thing that they have to do on the internet. Therefore, extensibility is important because although we have considered these cases, we may not have time to implement them, but that doesn?t mean that we will compromise the software so that it is forever static.

5. Key Development Technologies

The key development technologies include the integrated development environment, the continuous integration server, the testing framework, and the version control system.

Table **??** describes the key development technologies and the rationale behind the specified choice.

Eclipse was chosen as the IDE due to the fact that it is both open source and cross-platform. Eclipse has also been promoted in previous courses, so the majority of the group is already familiar with it. Unfortunately Eclipse does not have out of the box support for Ruby, so the plugin Aptana is being used for Ruby support.

Jenkins is being used as the CI server, and is being hosted here. Jenkins was chosen because it has an extremely large number of plugins and it is easy to extend. It also has support for many of the common version control systems including git. We also looked at CruiseControl.rb, but chose not to use it because it does not have a graphical reporting system. Git is being used as the version control system, and Trac is being used as the web interface to git. The git repository can be checked out using:

```
git clone <username>@seng403.ssh22.net:/var/cache/git/hotelmanager
```

The trac repository can be accessed here. We also tried the Gitlab web interface, but it was missing timeline and roadmap functionality and its bug reporting functionality was poor.

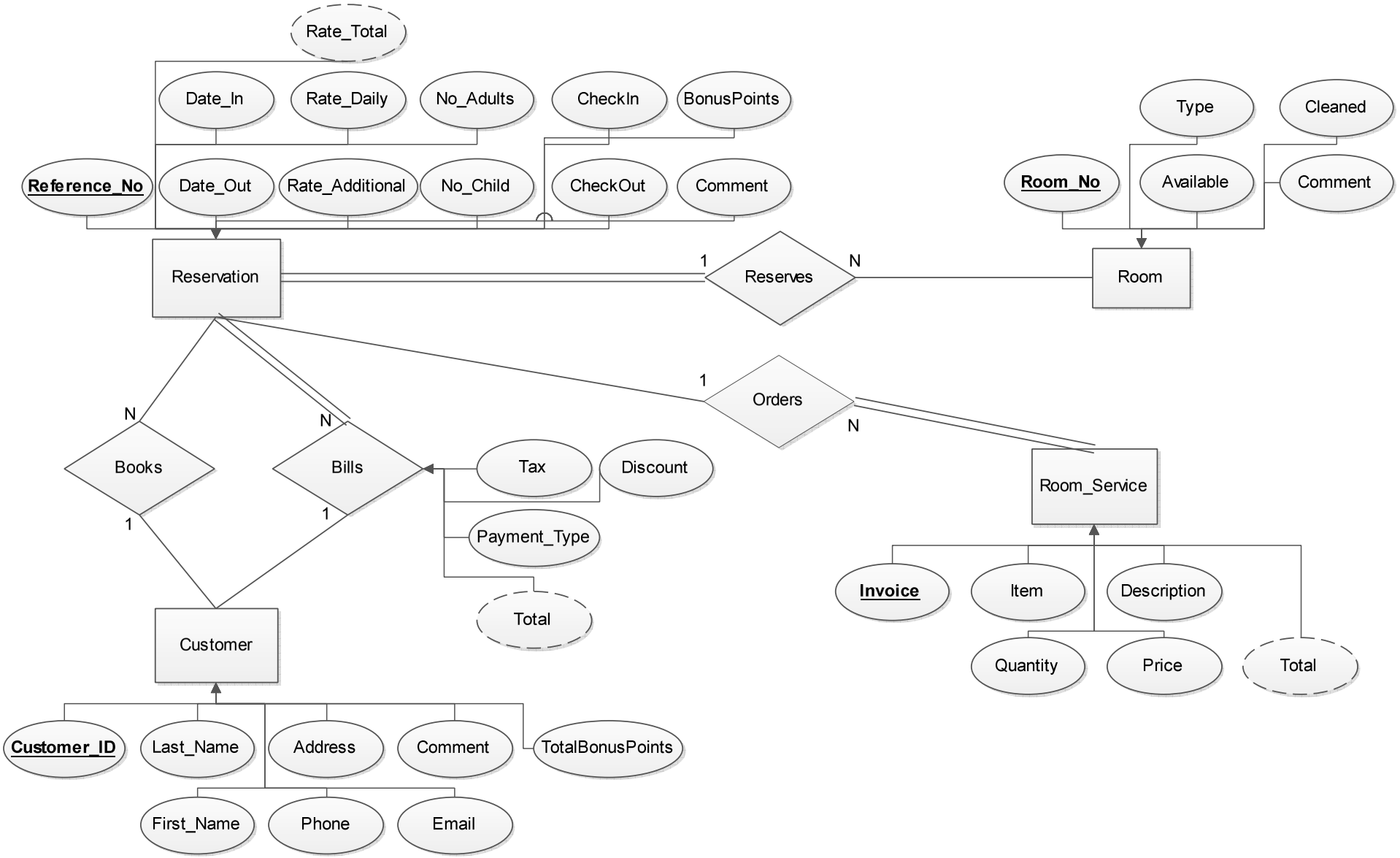
Shoulda will be used as the unit testing framework. This is because it is an extension of the built in Ruby unit testing framework Test::Unit. Unlike RSpec which encourages Behaviour Driven Development, Shoulda uses Test Driven Development which the group is more familiar with.

6. Critique

Name	Description
Brandon	Evaluated and setup continuous integration server, git, and trac Wrote the key development technologies section Re-wrote the release plan Re-wrote the operational requirements Setup Tex Document, with title page, etc Combined the report and produced the rough-draft
Julia	Wrote the critical success factors Chose the IDE plugin
Tyler	Wrote operational requirements
Garrick	Release plan Chose the IDE plugin
Alberto	Release plan

7. Appendix

Iteration	User Stories	Tasks	Difficulty/ Estimate	Status	Owner	Description
Iteration 0 Deadline Feb	Development Setup	system	Hard	Complete	Brandon	
		environment	Medium		Everyone	Install Aptana, Ruby, Rails.
Iteration 1 Deadline Feb 26	Administrator is able to maintain users	control	Hard			base user access control.
		administrator	Medium			specific behaviour.
		users	Medium			modify user accounts.
		User permissions	Hard			of other users and roles.
	Users are able to login to the system.	User control for sales associates	Easy			
		User control for hotel managers	Easy			
		User control for registered customers	Easy			
	Customer information management	Sales associates can manage customers	Hard			Sales associates will be able to add, delete, and modify customers.
		search for specific customers	Hard			Sales associates will be able to do a very simple search of the customers by the customer name.
Iteration 2 Deadline Mar 19	Hotel management	Administrators can manage hotels	Hard			Administrators will be able to add, delete, and modify hotels.
		hotels	Hard			Users will be able to search hotels
	Room management	Hotel managers can manage rooms	Medium			Hotel managers will be able to add, delete, and modify rooms.
		Hotel managers can manage room types	Easy			Hotel managers will be able to add, delete, and modify room types.
		Sales associates can view the list of rooms	Medium			Sales associates will be able to do a very simple search of the rooms.
	Room booking and management	manage room bookings	Hard			Sales associates will be able to add, delete, and modify room bookings.
		Sales associates can view bookings	Medium			Sales associates will be able to search for room bookings.
Iteration 3 Deadline Apr 9	Invoicing	record customer payments	Hard			Sales associates will be able to record customer payments and manage invoicing status.
		view customer billing history	Medium			Sales associates will be able to view customer billing history.
	Room service	service	Easy			
		view customers by room number	Medium			Room service associates will be able to view customers by room number.
		manage the room service bill	Medium			and modify the room service bill for customers at their hotel.
	Customer self-service	New customers can register to the system	Hard			Customers should be able to register with the system if they do not already have an account.
		customers	Easy			
		Customers can view available rooms	Easy			Registered customers will be able to view the available rooms.
		manage their room bookings	Easy			remove reservations, and view their current reservations and payment history.
Iteration 4 Deadline May 1	Trouble ticket management	Customers can issue trouble tickets	Hard			Customers should be able to issue trouble tickets which can be managed by hotel workers.
		Sales associates can resolve trouble tickets	Hard			trouble tickets, including responding and resolving issues.
	Housekeeping	housekeeping	Easy			
		manage cleanliness status	Hard			cleanliness status and mark rooms as cleaned, as well as view the rooms they are responsible for.
	Reporting	Hotel managers can produce reports on accounts receivable, hotel utilization, etc.	Hard			Hotel managers should be able to view and manage reports on the hotel's accounts receivable, view the hotel utilization, view the average rate where customers do not show up, view and suspend delinquent accounts, etc.


Rate_Total
Date_In
Rate_Daily
No_Adults
CheckIn
BonusPoints
Reference_No
Date_Out
Rate_Additional
No_Child
CheckOut
Comment
Reservation
1
Reserves
N
Room
Type
Cleaned
Room_No
Available
Comment
1
Orders
N
N
Books
N
Bills
1
1
Tax
Discount
Payment_Type
Total
Customer
Room_Service
Invoice
Item
Description
Quantity
Price
Total
Customer_ID
Last_Name
Address
Comment
TotalBonusPoints
First_Name
Phone
Email

8. References

[1] M. Maslanova. Summary/minutes for fesco 2013-01-30, 2013. [Online; accessed 5-February-2013].

[2] Wikipedia. Wikimedia traffic analysis report - browsers, 2012. [Online; accessed 5-February-2013].