GROUP DOCUMENTATION

Contents

[1 Abstract 4](#_Toc345628052)

[2 Introduction 5](#_Toc345628053)

[2.1 Client Overview 5](#_Toc345628054)

[2.2 Problem Statement 5](#_Toc345628055)

[2.2.1 Scope 5](#_Toc345628056)

[2.2.2 Problem 5](#_Toc345628057)

[2.3 Proposed Solution 6](#_Toc345628058)

[2.3.1 Goals 6](#_Toc345628059)

[3 Development Methodology 7](#_Toc345628060)

[3.1 Methodology Overview 7](#_Toc345628061)

[3.2 Implementation of Methodology 7](#_Toc345628062)

[3.2.1 Team and Roles 7](#_Toc345628063)

[3.2.2 Project Management 7](#_Toc345628064)

[3.2.3 Development Lifecycle 8](#_Toc345628065)

[3.2.4 Development 11](#_Toc345628066)

[3.2.5 Documentation 11](#_Toc345628067)

[4 Tools & Technologies 12](#_Toc345628068)

[4.1 Tools & Technology Overview 12](#_Toc345628069)

[4.1.1 Design Tools 12](#_Toc345628070)

[4.1.2 Development Tools 12](#_Toc345628071)

[4.1.3 Management and Communication Tools 12](#_Toc345628072)

[5 System Overview 13](#_Toc345628073)

[5.1 Requirements 13](#_Toc345628074)

[5.1.1 Functional Requirements 13](#_Toc345628075)

[5.1.2 Non-functional Requirements 14](#_Toc345628076)

[5.2 Reservations and Information Module 14](#_Toc345628077)

[5.2.1 Customer Focus 14](#_Toc345628078)

[5.2.2 Seat Reservations 15](#_Toc345628079)

[5.3 Reports and Intelligence Module 16](#_Toc345628080)

[6 Quality Assurance 18](#_Toc345628081)

[6.1 Quality Assurance Overview 18](#_Toc345628082)

[6.2 Test Plan 18](#_Toc345628083)

[7 Conclusion 19](#_Toc345628084)

[References 20](#_Toc345628085)

[Appendix A 22](#_Toc345628086)

[Problem Definition 22](#_Toc345628087)

[Needs and Vision 22](#_Toc345628088)

[Teams and Roles 23](#_Toc345628089)

[RAD Roles 23](#_Toc345628090)

[Scrum Roles 24](#_Toc345628091)

[Scrum Events and Artifacts 24](#_Toc345628092)

[Sprint 24](#_Toc345628093)

[User Story 24](#_Toc345628094)

[Product Backlog 24](#_Toc345628095)

[Sprint Planning 25](#_Toc345628096)

[Daily Scrum 25](#_Toc345628097)

[Sprint Retrospective 25](#_Toc345628098)

[Appendix B: Wireframes 26](#_Toc345628099)

[Appendix C: Entity-Relationship Diagram 28](#_Toc345628100)

[Appendix D: Tools and Technology 30](#_Toc345628101)

[Microsoft Team Foundation Services 30](#_Toc345628102)

[Google Services 30](#_Toc345628103)

[Appendix E: Solution Screenshots 32](#_Toc345628104)

# Abstract

# Introduction

## Client Overview

The client is modeled after Sri Lanka Railways (SLR), a government department under the Ministry of Transport of Sri Lanka. It provides railway services for passengers and freight, connecting Colombo with various locations in the island (Sri Lanka Railways, 2011). It is important to note that the client is only modeled after SLR, and SLR is not the client.

The client representative and senior management representative roles are the product owners and act on behalf of the modeled client. They specify the requirements, collaborate on product design aspects with developers and test each iteration of the product that is delivered.

## Problem Statement

### Scope

The scope is constrained to seat reservations on long distance passenger trains (hitherto referred to as *trains*) operated by SLR. It includes both SLR-operated and privately-operated coaches that are attached to these trains.

### Problem

The modus operandi of seat reservations (Government Information Center, 2011) are either by visiting the railway station during certain hours of the day, or using the M-ticket Reservation System operated by mobile phone operator Mobitel. Seats on privately-operated coaches have the additional option of reserving online.

The efficiency of existing reservation methods are not up to mark, and include the following problems:

* Cost, lost time and inconvenience travelling to the station to reserve seats, which moreover does not guarantee seats.
* Long lines at the ticket counter just before the train departs.
* The M-ticket option, requires customers to possess a mobile phone connection from Mobitel (Mobitel (Pvt) Ltd., n.d.).
* Preferred seats cannot be selected when reserving, except on online reservation of privately-owned coach seats.
* Train information is outdated, inconsistent and incomplete (Government Information Center, 2011).
* Reservation information not consolidated from various reservation methods.
* No means to analyze and gather meaningful insight from seat reservation data.

Refer section *Problem Definition* in *Appendix A* for elaboration

## Proposed Solution

An online Train Reservation System with Reporting and Business Intelligence was conceived as the solution. It contains three modules to attempt solve the problems discussed above; an *online seat reservation* module, a *train information* module and a *reporting and business intelligence* module.

### Goals

The solution aims to be the sole system that would be used for reservations by the public and other existing methods by enabling all types of bookable seats on SLR-operated and privately-operated coaches on all trains. It provides up to date and comprehensive train information, and enables stakeholders to analyze data and gain insight into their data for better business planning.

# Development Methodology

## Methodology Overview

We **adopted** the Manifesto for Agile Software Development (Beck, et al., 2001) as a guide to manage the development process, to suit the needs of the team and the vision the team had for the project. We found that the high value aspects described in the manifesto that are considered when developing software, and the principles that they are based on (Beck, et al., 2001) support our needs and vision rather well. Hence, we **adapted** Scrum, an agile software development methodology (Cohn, 2010) to suit our development needs.

Refer section *Needs and Vision* in *Appendix A* for elaboration.

## Implementation of Methodology

### Team and Roles

We designated a cross-functional team of individuals with multiple skill-sets, so that each member gets the opportunity and responsibility of working across several aspects of the project. This gives the members a sense of accountability, an opportunity to learn and contribute towards sustaining a self-organizing and cohesive team. The synergy between members collaborating on decisions and problem solving was expected to give the team a better probability at arriving at higher quality decisions.

Refer section *Teams and Roles* in *Appendix A* for elaboration.

Despite role assignments, each member was expected to perform responsibilities of other roles, especially development. Members were also expected to be proactive and communicate constantly such that problems and issues could be minimized.

### Project Management

Project and application lifecycle management tasks were performed by the facilitator while collaborating with the team. The project was managed on TFS using the Microsoft Visual Studio Scrum 2.1 template, with sprints time-boxed to a week each.

#### Product Backlog Management

We primarily used a combination of MoSCoW analysis (International Institute of Business Analysis, 2009) and basic principles of the minimum marketable features (Denne & Cleland-Huang, 2004) in prioritizing user stories. Each user story included a Definition of Done (DoD). This was to help the team have a shared understanding of what it means for an item in the Product Backlog to be complete and assess it when it is completed (Schwaber & Sutherland, 2011).

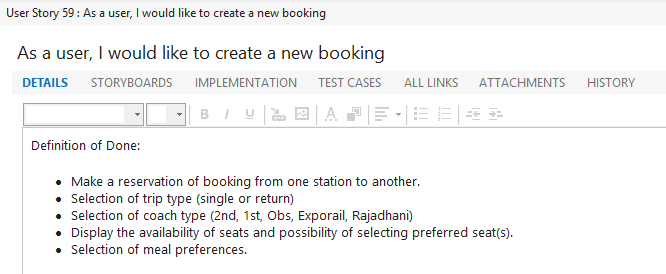


Figure 1: A User Story with Definition of Done from the TFS Product Backlog

User stories were then broken down into workable tasks. Next, the development team collaboratively discussed and assigned a weight based on the magnitude and complexity of the task. Using a combination of the above measurements we sorted and grouped the tasks into sprints.

#### Scrum Activities

Each sprint started with sprint planning to ensure that the team is on the same page and know what is expected as output at the end of it. Since communication was an important aspect in an agile team (Beck, et al., 2001), the development team held daily scrum meetings on a daily if not regular basis, that were time-boxed to 20 minutes. Sprint retrospectives, time-boxed to 15 minutes were held at the end of each sprint to reflect upon work done during the spring and set plans for improvement.

### Development Lifecycle

Development took place in week-long sprints including Sprint 0, which was set aside to system study and initial requirements gathering.

#### Sprint 0

The initial sprint (Sprint 0) was a phase of studying and understanding requirements, and discussing and collaborating on ideas between the product owner and the developer team. Discussions started with the management representative stating their problems, and the user representative coming up with probable solutions. The developer team then begins furnishing ideas for the solution, which everyone collaborate upon. The inferred user stories were added to the Product Backlog in TFS. Higher priority backlog items would be more detailed than the emergent ones.

#### Disposable Prototyping

Our strategy on prototyping was implemented by developing disposable wireframes. These wireframes acted as the initial preview of the system to the client. As new user stories were introduced, a corresponding task was added to the product backlog for wireframe creation. During Sprint 0, the team prioritized on creating wireframes first (see Figure 2). These were then shown to the user representative for feedback. As we received feedback, the wireframes were revised with the changes and new tasks were added to the product backlog to implement these changes. The wireframes doubled as a user interface guide for the development team. As the development progressed through the iterations, the working prototype system gradually took on the center stage during the demos and the wireframes were no longer required.

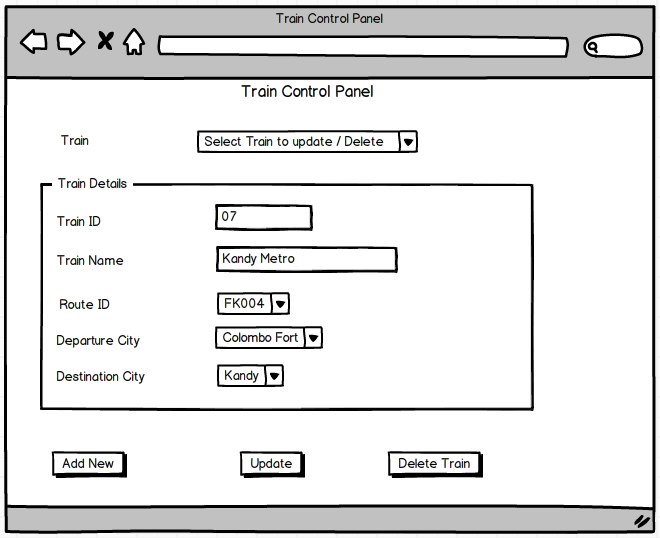


Figure 2: Wireframe of Train Management screen

It is important to note that documentation was limited to user stories and wireframes only. And these were only created for limited to complex functionality. Documentation was traded in favor of functionality that would give the client more value.

#### Sprints 1 to 5

Sprints 1 to 5 were designated as development sprints, and began with a sprint planning session. As the team progressed through the backlog items and implementation became concrete, tasks were detailed out with additional information. This gave the team more current information at the time of development as well as serving as additional documentation. Once development and acceptance testing for the sprint ends, the team convenes for a sprint retrospective.

#### Sprint 6

Once sprints 1 to 5 were completed, the client brought in new and extensive requirements in the form of reporting and management information modules. Hence, Sprint 6 was time-boxed to a month in order to deal with the significant amount of development, and the project deadline increased. This sprint was originally conceived as a place holder to pick up any tasks that were trailing from the previous sprints.

### Development

We used a model similar to one suggested by (VersionOne, Inc., 2013) to come up with a development velocity of the initial sprint; we used one third of the total ideal time available for the individuals. After the completion of the first sprint we used historical information from the previous ones to adjust the number of user stories for the iterations.

Team members picked out items assigned to them from the Product Backlog on TFS, updating the items as they progressed through their tasks. We observed instances of Pair Programming (Williams & Kessler, 2003) happen when the development team convened to develop together. When each member worked alone, it was observed that continuous interaction and support happening online.

### Documentation

Documentation for development was kept at a minimal. The only documentation that were used were the user stories in TFS, the wireframes (See *Appendix B: Wireframes*) and the entity-relationship diagram (See *Appendix C: Entity-Relationship Diagram*).

A weblog was maintained by the scribe for day-to-day activities, including project management, development tasks and communication. See for elaboration.

# Tools & Technologies

## Tools & Technology Overview

Technology was chosen based on the skill make-up of the team, the efficiency of the technologies in RAD, and the learning curve involved in learning new tools and methodologies. Strong emphasis was placed on tools that encourage and facilitate rapid application development. Infrastructure, environmental requirements, and the cost and easy availability of the tools also played a big part.

### Design Tools

* Balsamiq Mockups 2.2.3 (Trial Version)
* Microsoft Visio 2013

### Development Tools

* Microsoft Visual Studio 2012 Express for Web
  + .NET Framework 4.5 and ASP.NET using
    - Visual C#
    - ASP.NET MVC 4 (Model-View-Controller) pattern
  + Entity Framework 5
    - Code-First
* Microsoft SQL Server 2012 Express Edition with Advanced Services
* Microsoft SQL Server 2012 Enterprise Edition (90-day trial)
  + Analysis Services (Tabular)
* Microsoft Excel 2013

### Management and Communication Tools

* Microsoft Team Foundation Services (TFS Online)
* Google Services (including Google+, Google Calendar, Gmail, Google Talk, Blogger)

Refer *Appendix D: Tools and Technology* for elaboration and justification

# System Overview

## Requirements

### Functional Requirements

Functional requirements were not completely visible at the inception of the project. When we progressed through the iterations and collaborated with the stakeholders more, the functional requirements took proper shape.

* A potential customer should be able to view scheduled trains and fares
* A customer should be able to make a booking in the system without having to register or log into the Online Reservation System. They should be able to:
  + View a list of train schedules for the select route.
  + Select a preferred coach
  + Make individual selections for seat and meal preferences for each of the passengers.
* A customer should be able to register with the system and save his profile information
* A registered customer should be able to
  + View a summary of his previous bookings
  + Cancel his previously booked tickets
  + Re-print his tickets
  + Edit his profile information
* An administrator should be able to
  + Manage Coach Information
  + Manage Trains and Station Information
  + Manage Routes and Schedules
  + Manage Fares for the Routes
  + Edit and Update User Profile Information for all the users registered in the system.
* An administrator should be able to view the following reports
  + Dashboard
  + Seats Available
  + Destination-wise Sales Report
  + Train-wise Sales Report

### Non-functional Requirements

* User friendliness (fancy GUIs).
  + Well defined interfaces.
  + Ease of use and learning of the system.
* Portability and Robustness of the system
  + Browser and device compatibility.
* Security aspects of the system.
* Data backup and recovery mechanisms.
* Accessibility of the system (how many customers can access the system in given time).
* Performance of the system.
  + Response time.
  + Processing time.
  + Report generation time.
* Scalability of the system.
* Availability of the System.

The software solution is segregated into two modules: *Reservations and Information* and *Reports and Intelligence*.

## Reservations and Information Module

The Reservations and Information module is a web application that focuses primarily on travelers who wish to travel long distance on trains. Additionally it provides functionality to administrators of the system to help manage the entities that are part of the system.

### Customer Focus

Elaborating on the traveler or customer focus of the system, foremost is Train Information, where anyone visiting the web site can easily gain access to train information before making reservations. Train information is displayed on the home page of the website, providing service information including origin and destination, departure and arrival times, travel duration, train name and fare information. The information widget is completely searchable and provides the customer quick access to various information aspects of their train journey.

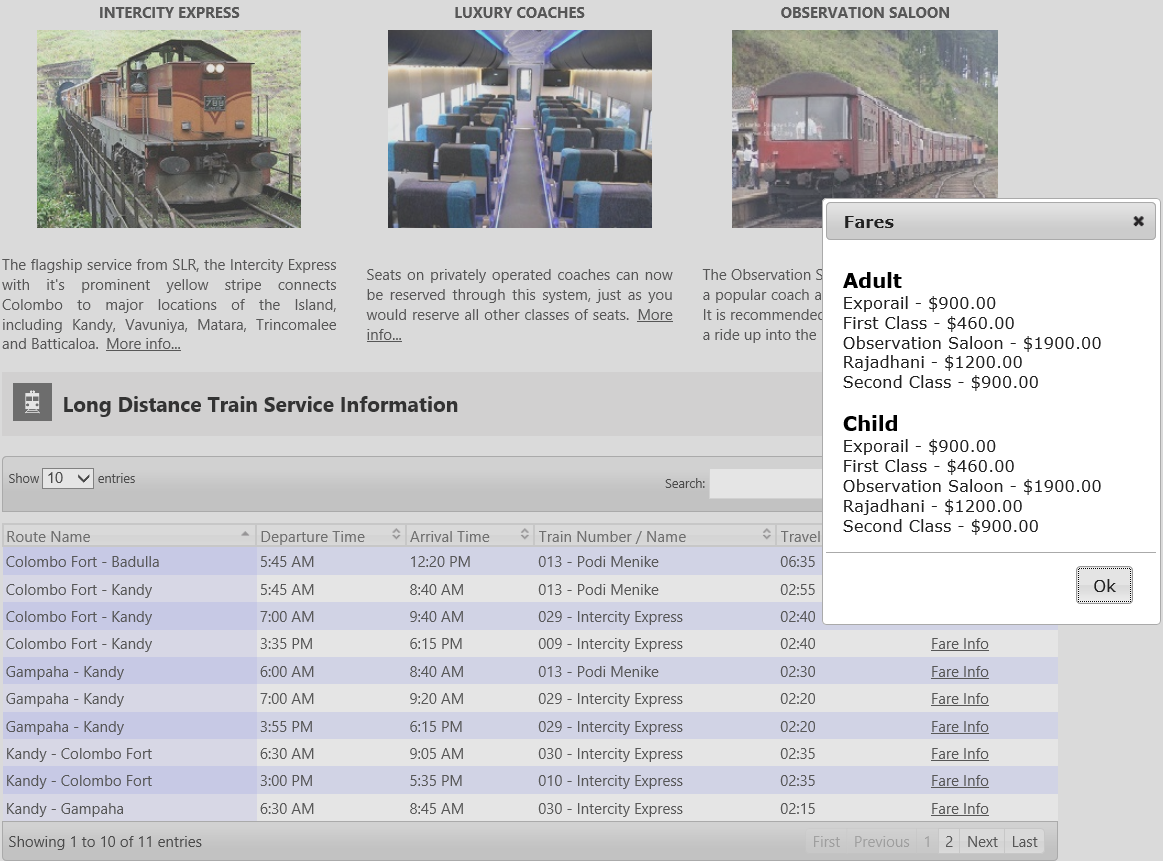


Figure 3: Searchable train information on the home page

### Seat Reservations

Additionally, it provides comprehensive information about long distance train services and the means to reserve seats on these trains. It also caters to designated SLR employees to perform management tasks on the system. The following is a feature break-down of the module with descriptions:

* **Train Information Section**: Displays train service information including origin and destination, departure and arrival times, travel duration, train name and fare information. This widget is completely searchable and provides the customer quick access to various aspects of their train journey.
* **Customer Dashboard**: Provides a list of services to the customer that they can utilize on the website. It is the main portal from which the customer gains access to their services.
* **Reservation Management**: Provides mechanisms to reserve, pay for and print tickets for a journey, cancel previously booked tickets, re-print previously booked tickets and view the history of ticket reservations performed by the customer.
* **Profile Management**: Provides the customer with the facility to register with the system and maintain their personal profile.
* **Administrator Dashboard**: Provides a list of services to SLR employees who are designated as administrators of the system. It is the portal from which the administrator gains access to their services.
* **Entity Management**: Provides functionality for administrators to manage the various entities that are part of the system. These include the following activities:
  + Creating trains, coaches and seats and allocating them appropriately and managing them.
  + Creating stations, routes and schedules and allocating trains to them.
  + Managing fares for the different seats for the different schedules and routes.
* **User Profile Management**: Enables administrators to manage information of registered information

Refer to Appendix for Reservations and Information module screenshots

## Reports and Intelligence Module

This module provides the management and administrators (collectively called business users) perspective to the information captured by the system.

The aim of this module is to provide business metrics and knowledge to the stakeholders that would empower them to make better business decisions based on these information. This module provides insight through traditional static reports and dashboards.

As well it provides analytic information that can be visualized and analyzed in dynamic ways using every day tools such as Microsoft Excel. The module enables this by providing the business user with a data model which the business user can exploit to perform analysis through a concept known as self-service business intelligence. Self-service business intelligence gives business users the ability to create small-scale BI solutions that require little or no help from IT departments. It makes these solutions extremely business-focused and fast to build (Russo, et al., 2012).

Refer to Appendix for Reports and Intelligence module screenshots

# Quality Assurance

## Quality Assurance Overview

Our testing strategy for the project was based on the work carried out during the time span of a sprint. The Test plans were created in the Team Foundation Server each sprint and tested at the end of the sprint against a deliverable user story. Each team member took the responsibility of creating the Test Cases against the user story he was working on and completing the test as the development work was completed. This was done in the true spirit of Agile where test drive development is emphasized as it cuts down bugs since the testing is carried out by the developer who did the actual coding.

## Test Plan

# Conclusion

# References

Beck, K. et al., 2001. *Manifesto for Agile Software Development.* [Online]   
Available at: http://agilemanifesto.org/  
[Accessed 28 December 2012].

Beck, K. et al., 2001. *Principles behind the Agile Manifesto.* [Online]   
Available at: http://agilemanifesto.org/principles.html  
[Accessed 28 December 2012].

Blue Line Company, 2011. *http://www.rajadhani.lk.* [Online]   
Available at: http://www.rajadhani.lk  
[Accessed 4 January 2013].

Cohn, M., 2004. An Overview. In: *User Stories Applied: For Agile Software Development.* s.l.:Addison-Wesley Professional, p. 4.

Cohn, M., 2010. Why Becoming Agile Is Hard (But Worth It). In: *Succeeding with Agile: Software Development Using Scrum.* s.l.:Addison-Wesley Professional, p. 4.

Denne, M. & Cleland-Huang, J., 2004. The Incremental Funding Method - A Data Driven Approach to Software Development. *IEEE Computer Society,* 21(3), pp. 39-47.

Exporail, 2013. *http://www.exporail.lk/.* [Online]   
Available at: http://www.exporail.lk/  
[Accessed 4 January 2013].

Government Information Center, 2011. *Reservation of Seats in Intercity and other Express Trains.* [Online]   
Available at: http://www.gic.gov.lk/gic/index.php?option=com\_info&task=info&id=432&lang=en  
[Accessed 5 January 2013].

International Institute of Business Analysis, 2009. 6.1.5 Techniques. In: K. Brennan, ed. *A guide to the Business Analysis Body of Knowledge (BABOK guide), version 2.0.* s.l.:IIBA, p. 102.

Mobitel (Pvt) Ltd., n.d. *Mobitel - Ticketing.* [Online]   
Available at: http://www.mobitel.lk/ticketing  
[Accessed 5 January 2013].

Russo, M., Ferrari, A. & Webb, C., 2012. Self-Service BI and Corporate BI. In: *Microsoft SQL Server 2012 Analysis Services: The BISM Tabular Model.* s.l.:O’Reilly Media, Inc., pp. 4-5.

Schwaber, K. & Sutherland, J., 2011. *The Scrum Guide,* s.l.: s.n.

Sri Lanka Railways, 2011. *Sri Lanka Railways.* [Online]   
Available at: www.railway.gov.lk  
[Accessed 1 January 2013].

VersionOne, Inc., 2013. *Measuring the Velocity of your Scrum Team.* [Online]   
Available at: http://www.versionone.com/Agile101/Agile-Scrum-Velocity/  
[Accessed 6 January 2013].

Williams, L. & Kessler, R. R., 2003. Introduction. In: *Pair Programming Illuminated.* s.l.:Addison-Wesley Professional, p. 3.

# Appendix A

## Problem Definition

One relatively lesser problem is, train information available to a traveler being outdated, inconsistent and incomplete. Any traveler, before embarking or reserving seats for a journey would need to know information about the journey, such as type of train (whether express or slow), the classes of seats available, departure and arrival schedules, meal options etc. This information is not consistently available in a proper consolidated format, even on the official government information website (Government Information Center, 2011) which officially supplies this information (Sri Lanka Railways, 2011).

Stakeholders are looking at increasing their market share in passenger transport beyond the 6% (Sri Lanka Railways, 2011) which they currently enjoy. However, there are no means to gather intelligence that would allow them to appropriately plan their sales activities. Hence, the stakeholders require means of analyzing reservation information and obtain meaningful insight from it.

## Needs and Vision

For a dynamic development project of short duration, the team had the following needs and vision for the project:

* Concentrate primarily on delivering quality functionality and value to the stakeholders.
* Encourage the stakeholders to focus on what is important in terms of value.
* Encourage the client to come up with new ideas and collaborate with the team to get a competitive edge in the business.
* Deliver robust modular software in a very short period.
* Undertake change requests and direction changes from the client even at the final stages of development.
* Adopt and integrate the latest technologies and tool and ensure that the developed solution is reasonably future proof.

## Teams and Roles

One each of the five RAD roles were assigned to each member, who performed role-specific responsibilities and also assuming additional responsibilities, even of other roles as and when needed. Development of the solution was however expected to be performed by each member.

### RAD Roles

#### Facilitator

Shane Carvalho was chosen for this role based on his recent training on agile project management techniques and several years' worth of team management in the capacity of a technical lead. This role’s functionality includes playing the Scrum Master and technical facilitations akin to that of a technical lead.

#### Management Representative

Upul Weeramuni was chosen for this role based on his understanding and experience drawn from maintaining railway reservation systems. He were deemed to prove beneficial in understanding and eliciting requirements from similar systems their organization deals with. This role drives the vision and the mission of the client. This role will define the user stories with the assistance of the User Representative.

#### User Representative

Anuradha Thilakarathne was handed this role based on interest and requirements engineering experiences. This is the role of the product owner, which bridges the technical role(s) to the management role(s). The role also collaborates with the client management role and provides value/weight to the user stories.

#### Developer

Chosen based on his experiences and expertise in development and design patterns, Ahsan Sally was delegated to lead the development of the solution prototype, including leading architectural and design aspects of the solution.

#### Scribe

Gogula Aryalingam was chosen for this role centered on experiences of technical documentation and technical writing on previous team projects. He was considered the ideal person to keep regular journal entries and document details of the project.

### Scrum Roles

The Scrum roles of Scrum Master and Product Owner, were primarily played by the facilitator and the user representative respectively. In their absence these Scrum roles were played by the other roles of the team.

#### Scrum Master

The Scrum Master ensures scrum is practiced by the team and facilitate interactions within it such that it maximizes the value produced by the team. The Scrum Master manages the product backlogs, facilitates communication and helps with removing impediments to the team’s progress (Schwaber & Sutherland, 2011). This role was played primarily by the facilitator and by the scribe and occasionally by the developer in the absence of the facilitator.

#### Product Owner

## Scrum Events and Artifacts

The following elaborate how we adopted Scrum to fit our needs:

### Sprint

A Sprint is a time-boxed unit within which a useable and releasable product is created by the development team (Schwaber & Sutherland, 2011).

### User Story

A User Story (Cohn, 2004) describes functionality of a system or software that is required by a user. It is used to document details such that when the functionality is built, this description can be used to evaluate its completeness (Cohn, 2004).

### Product Backlog

A Product Backlog is an ordered list of everything that the product needs to have. It is considered to be the repository of requirements, features and enhancements (Schwaber & Sutherland, 2011).

### Sprint Planning

Sprint Planning occurs at the beginning of each sprint and the whole team is involved, headed by the Scrum Master. Here the team performs a walkthrough of the product backlog to ensure that each role is on the same page before development begins. The product backlog gets updated with more details during sprint planning as new ideas and functionality get added and the team understands aspects better.

### Daily Scrum

Daily scrums (Schwaber & Sutherland, 2011) allows each member to be in sync with where the team and project stands, while also helping to identify impediments faced by the team, so that it could be resolved faster. The facilitator functions as the Scrum Master, with the scribe or the developer playing the role in their absence. Each member of the team explains the following during the daily scrum:

* What they did since the previous meeting
* What will be done before the next meeting
* What impediments that are blocking their progress

### Sprint Retrospective

At the end of each sprint, the team held a sprint retrospective time-boxed to 15 minutes. This (Schwaber & Sutherland, 2011), was in order to give an opportunity to the team to reflect upon our performances with regards to the various aspects of the sprint. The team identifies what went well in the sprint, and what needed improvement. Improvements for the team to better perform their work was documented for the next sprint.

# Appendix B: Wireframes

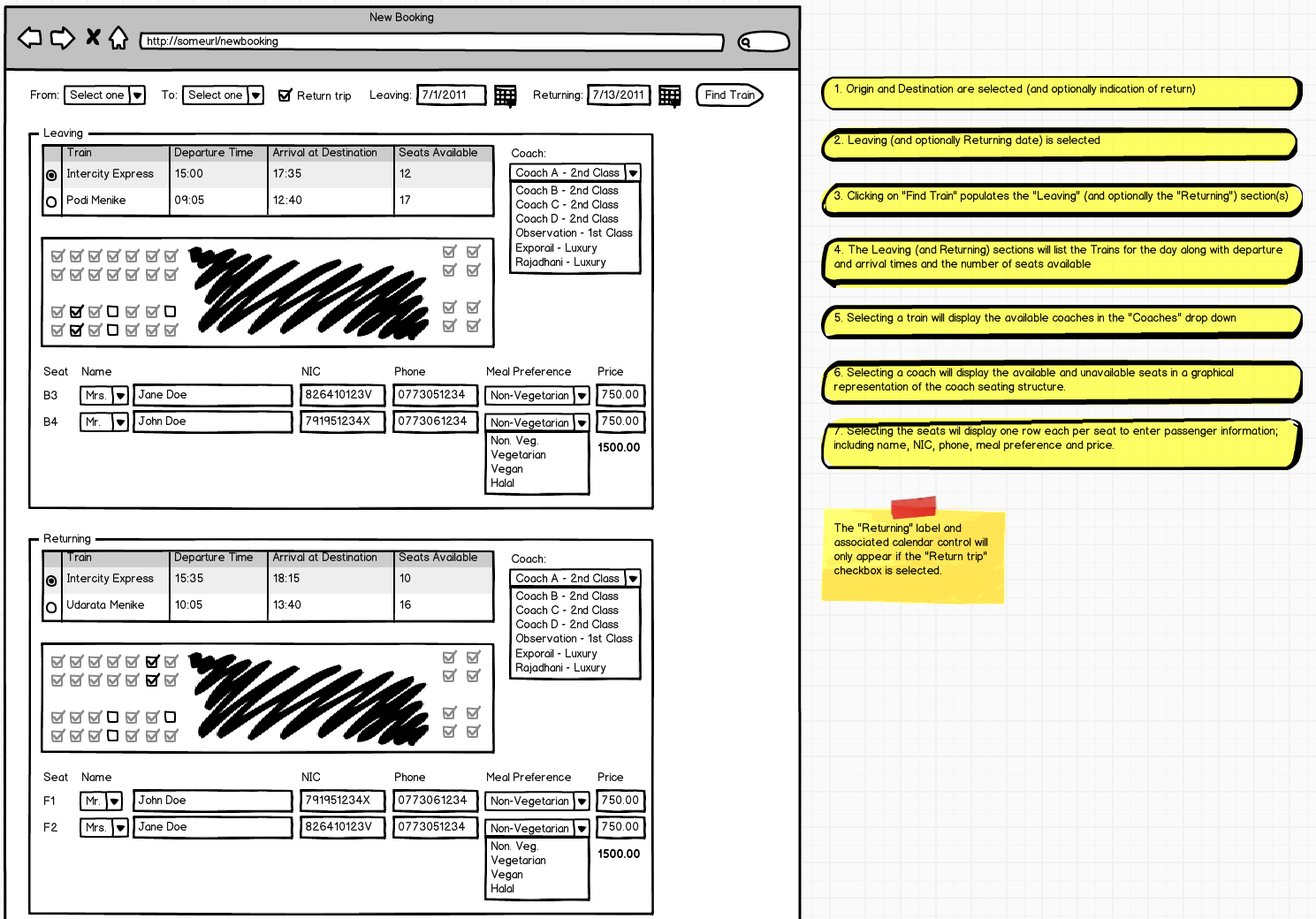


Figure 4: New Seat Reservation wireframe with notes

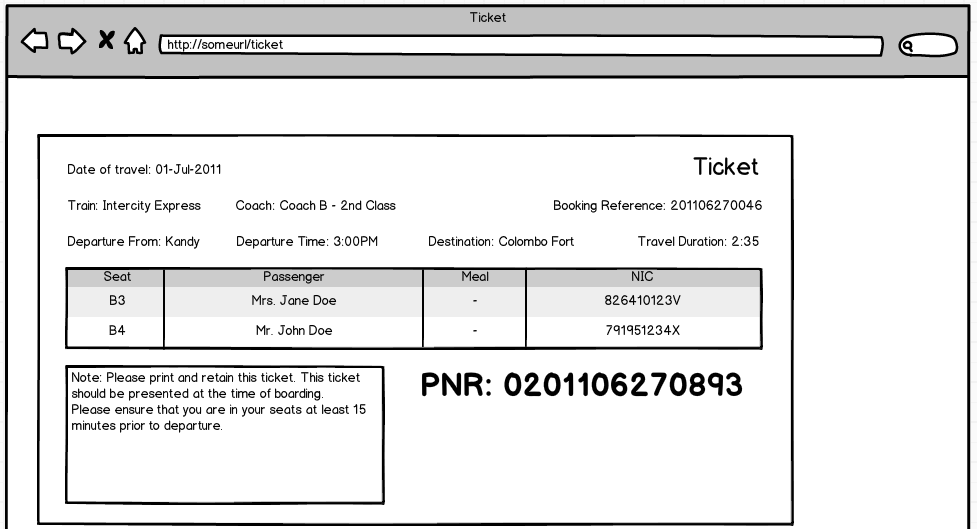


Figure 5: Ticket wireframe

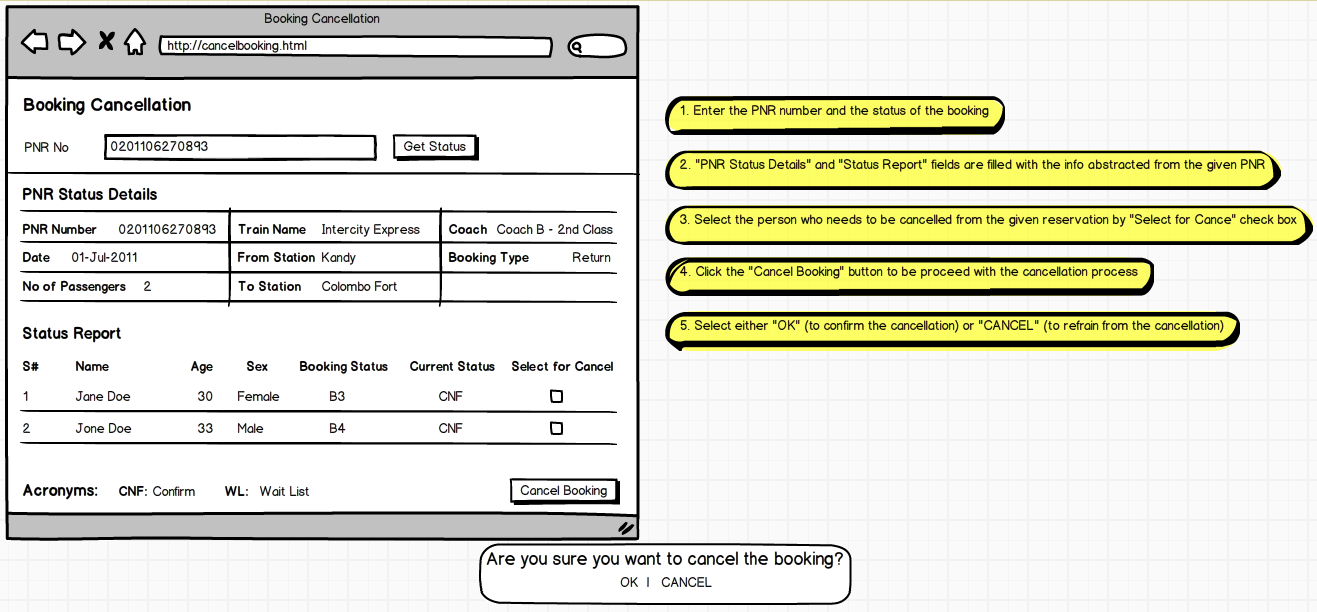


Figure 6: Seat Reservation Cancellation wireframe with notes

# Appendix C: Entity-Relationship Diagram

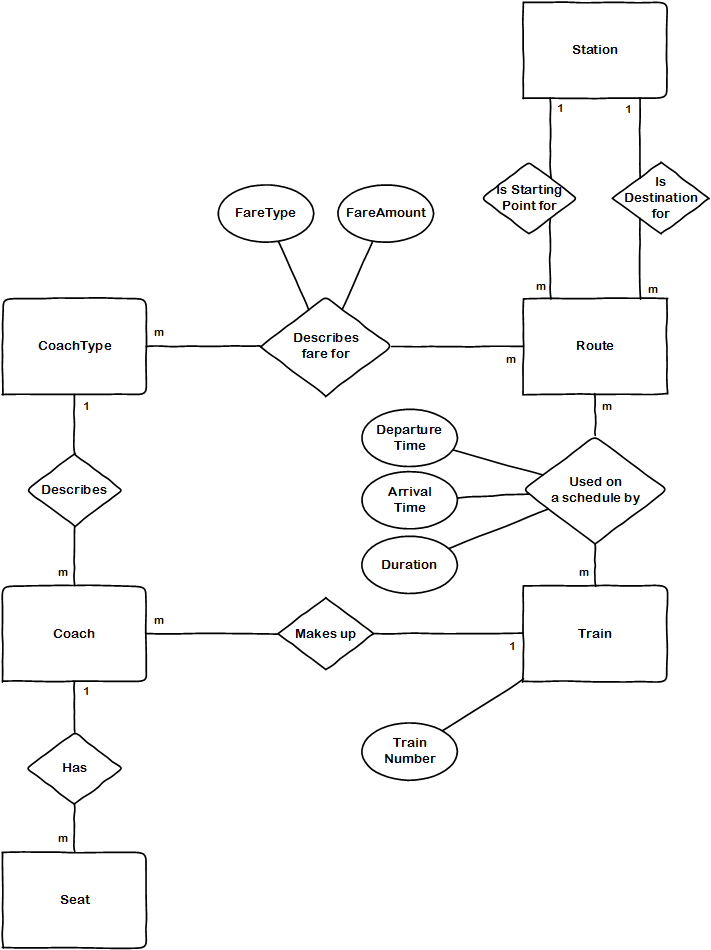


Figure 7: Entity-Relationships of main entities

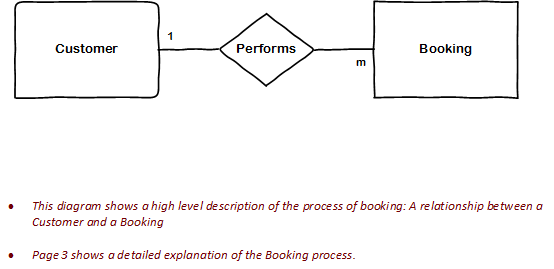


Figure 8: Entity-Relationship of Booking (High Level)

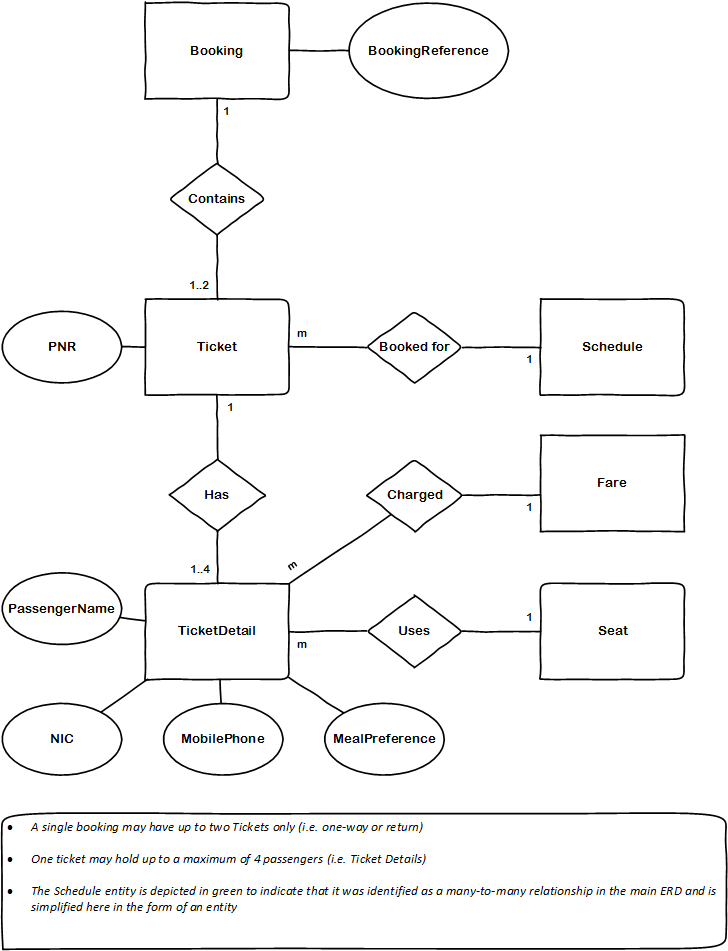


Figure 9: Entity-Relationship of Booking (Low-Level)

# Appendix D: Tools and Technology

## Microsoft Team Foundation Services

Microsoft Team Foundation Services (TFS) was adopted to perform project management, artifact management and source control activities. This service is an online service of Microsoft’s Team Foundation Server application lifecycle management software. It provides the following features that were used in the project:

* Out-of-the-Box Agile Templates
* Source Control
* Collaboration Framework
* Bug Tracking
* Reports

Since it is hosted in the cloud there were no infrastructure and setup cost considerations that had to be taken into account. Seamless integration with products within the Microsoft ecosystem such as Visual Studio and Excel made working with it trouble free. The service is provided free for small teams of up to five until the beginning of the following year (2013), which was ideal for the team size and adequate for the short duration of the project. The Agile templates aided the team in quickly setting up user stories, tasks and related artifacts and focus on the actual work at hand, while providing high visibility on the progress of the various project aspects.

## Google Services

Various services provided by Google were used for communication and journaling. Services and tools used:

* Google+
* Google Calendar
* Google Mail
* Google Talk
* Blogger

Communication was primarily carried out via Google+ and in ad-hoc occasions using Google Talk. Documentation to maintain a historical timeline of the project progress was performed through a weblog created at Google’s blogger.com.

Alternate tools that were considered included Facebook, Skype, basecamp.com and a locally hosted SharePoint site. The decision to use Google’s various services was based on the unique feature set offered:

* Built-in video conferencing offered via Google+ Hangouts.
* All team member already had Google accounts and were familiar with the intuitive interface resulting in a low learning curve.
* Announcements are easily communicated in a timely manner via the Google Hangout Posts.

Scrum meeting appointments can be integrated directly into Google Calendar.

# Appendix E: Solution Screenshots