MXC-0301

WPI Helpdesk Employee Management System

A Major Qualifying Project Report
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Bachelor of Science
in Computer Science

Jason Codding
Eric Greer
4/29/2010

Advisor

Michael Ciaraldi

Abstract

The goal of this project was to design and develop an employee management system for the WPI Helpdesk. This system was developed to fill existing gaps in the electronic management of employees. The scope of the project included researching and evaluating various Web technologies and analysis and design of employee profiles, electronic timesheets, employee performance reviews, and an integrated hiring process. Development work included construction of the core system framework, creation and management of accounts, employee profiles, and electronic timesheets, and a foundation for future modules.

Table of Contents

Abstract i

Table of Contents ii

Table of Figures iv

Table of Tables iv

1 Introduction 1

2 Background 2

2.1 Business Need 2

2.1.1 Project Focus Shift 2

2.1.2 Functional Requirements 2

2.1.3 Manager Functionality: 4

2.1.4 Employee Functionality: 5

2.1.5 Nonfunctional Requirements: 5

2.2 Technology Exploration 6

3 System Design 10

3.1 System Functionality 10

3.1.1 Scenarios 10

3.1.2 Use Cases 10

3.1.3 Data Model Diagrams 12

3.1.4 Prioritization of Features 14

3.2 User Interface 16

4 Implementation 18

4.1 Technology 18

4.1.1 Google App Engine 18

4.1.2 EXT GWT 19

4.2 System Components 20

4.2.1 Login/User Accounts 20

4.2.2 User Profile 21

4.2.3 Timesheets 23

5 Future Development 26

5.1 Deployment Instructions 26

5.2 Know Issues 26

5.3 Additional Functionality 27
Appendix A: Scenarios 28
Appendix B: Use Cases 31
Appendix C: Screenshots 40
Appendix D: Timeline 42
Appendix E: Detailed Data Requirements 43

Table of Figures

Figure 1: User Profile Data Model Diagram 12

Figure 2: Timesheet Data Model Diagram 12

Figure 3: Review Data Model Diagram 13

Figure 4: Hiring Data Model Diagram 13

Figure 5: System Architecture 20

Figure 6: Login Form 20

Figure 7: Account Listing 21

Figure 8: Employee Profile From 22

Figure 9: Position Management 23

Figure 10: Employee Profile Listing 23

Figure 11: Manager Timesheet Listing 24

Figure 12: Timesheet View 25

Figure 13: Timesheet Details 25

Figure 14: Example PDF Timesheet 40

Figure 15: Position Listing 40

Figure 16: Edit Position Form 41

Figure 17: Change Password Form 41

Table of Tables

Table 1: Technology Comparison 8

Table 2: Project Timeline 42

1 Introduction

The goal of this project was to work with the WPI Helpdesk to automate processes and improve efficiency for the organization. This was accomplished by creating a Web based employee management system. While discussions began around the desire to improve the scheduling algorithm the project focus evolved into the creation of a complete system for managing many data maintenance and workflow operations. Those employee management tasks that we selected to integrate in the system included: employee profiles, electronic timesheets, employee performance reviews, and the hiring process. Early phases of the project began with the research and evaluation of a variety of Web technologies and comparing their capabilities with our system, security, functional, and user interface requirements. Application development work included working with Google App Engine, Google Web Toolkit, and the EXT-GWT Extension Library. Our analysis, design, and implementation procedures and outcomes are detailed throughout this document.

2 Background

2.1 Business Need

Since our freshman year (2006-2007) both project partners have been WPI Helpdesk employees. This allowed us to witness firsthand the difficulties of managing a student employee workforce. One of the issues that arose every year was creating a schedule for all employees that met their preference and course schedule requirements but also provided an adequate and fair delegation of hours. Previous implementations of scheduling systems included CGI based HTML forms, Excel spreadsheets, and shared Outlook calendars. Thus, when project discussions first began the intent was to create a new Website based system that would provide increased functionality and better schedule management capabilities for the Helpdesk manager. It was also about this same time that the Helpdesk began testing yet another new scheduling system called WhenToWork[1].

2.1.1 Project Focus Shift

As project discussions continued with WPI Helpdesk student staff and management it became evident that the WhenToWork system was providing a useful algorithm and interface for the scheduling of employee hours. However, these discussions also exposed a greater need: the ability to easily complete many of the repetitive employee management tasks. There was no simple and universal way for keeping track of all user contact details, their employee ranking, user responses to solutions, and general comments. These discussions and further analyses of the business need led to the design of what we now refer to in full as the 'WPI Helpdesk Employee Management System'.

2.1.2 Functional Requirements

The following sets of functional requirements were created in cooperation with Helpdesk management and student staff. These requirements describe the initial business processes that were to be handled by the Employee Management System. The requirements were categorized into the system

[1] WhenToWork is on online scheduling system. Details on this product can be found on their Website at http://whentowork.com/

components that reflect the primary tasks associated with employee management. Additionally, more detailed data requirements can be found in Appendix E: Detailed Data Requirements.

2.1.2.1 User Profile:

The user profile contains the employee's biographical, employment, and positions details. The user profile is described in full detail in section 4.2.2 User Profile.

- Allow multiple 5 star tickets[2]
- Allow multiple social networks
- Create an user profile from an application

2.1.2.2 Timesheets:

The Timesheets component processes the bi-weekly payroll for all Helpdesk employees. It is described in full detail in 4.2.3 Timesheets.

- Pull data from the WhenToWork schedule to generate timesheet
- Printing super scripts, and notes with any special cases
- Export to a universally printable object
- Generate default time sheet for each employee
- Auto assign new timesheets to employees
- Ability to create blank timesheet
- Calculate gross pay

2.1.2.3 Performance Reviews:

The performance reviews are an annual questionnaire process undertaken by the WPI Helpdesk Manager to evaluate and provide feedback for each employee. This component establishes an electronic means of creating the questionnaires, collecting the responses, and storing them for future reference.

- Create reviews based on of existing templates
- Bulk assign review to employees

[2] Tickets that receive five stars (the highest rating) on the customer feedback survey

2.1.2.4 Hiring Process:

The hiring process is the procedure through which a perspective employee applies to the Helpdesk via a Web application. The manager is then able to retrieve and review applications before accepting or denying their employment. The manager may also add comments and interview notes for each applicant.

- Display visible templates
- Notify manger of new applications
- CAPTCHA[3] protection

2.1.3 Manager Functionality:

2.1.3.1 User Profile:

- Has exclusive edit privileges on an Employee's:
  - Name
  - Gender
  - Date of Birth
  - Federal Work Study vs. Not
  - Hire Date
  - Ranking
  - Ranking Comments
  - Goals
  - 5-Star Tickets
- Add Positions for Helpdesk workers
- Modify Positions for Helpdesk workers
- Ability to manually create user profile

2.1.3.2 Time Sheets:

- Approval time sheets
- Ability to print default/blank timesheets for individuals
- Archive timesheets for a pay period
- Print all time sheets
- Print select time sheets
- Mark timesheets as printed

2.1.3.3 Performance Reviews:

- Create Templates
- Modify Templates

[3] CAPTCHA is a robot prevention measure used on Web forms that requires users to enter a word displayed in an image before submitting. More details can be found at http://captcha.net

- Save Templates
- Finalize Template
- Ability to bulk issue user reviews
- Set visibility on Manager reviews
- Save In Progress Reviews
- Complete Outstanding Reviews
- View User's reviews
- View Users who have not Completed their User Review

2.1.3.4 *Hiring Process:*

- Receive Notification of new applications
- Create Sticky Notes
- Automatic Job Offer/Rejection notifications as Helpdesk@wpi.edu
- Delete an Application

2.1.4 Employee Functionality:

2.1.4.1 *User Profile:*

- Does not have edit privileges on own:
  - Name
  - Gender
  - Federal Work Study vs. Not
  - Hire Date
  - Ranking
  - Ranking Comments
  - Goals
- Can set the initial values of:
  - Gender
  - Date of Birth

2.1.4.2 *Time Sheets:*

- Ability to add/edit/remove special cases
- Ability to submit timesheet

2.1.4.3 *Performance Reviews:*

- View Review History
- Save In Progress Reviews
- Complete Outstanding Reviews

2.1.5 Nonfunctional Requirements:

- Supports Roughly 50 Users
- Protects personal data
- 99% up time

- Cross browser compatibility
- Logging
    - Timesheets
    - Hiring
- Backups
- System must be well documented
    - Code
    - User Functions
- Helpdesk Maintainable server

2.2 Technology Exploration

Upon completion of the requirements for the application we began to review the many Web technologies that were available to us. We soon discovered that the days of simple HTML and JavaScript were long over and a plethora of competing technologies presented themselves as viable options. This provided an opportunity to research the benefits and drawbacks of each technology and determine which might be the most viable and appropriate solution for our project. Our first step was to catalog the positives and negatives of each technology to determine what role they could play in our application. A complete listing of these details can be found in Table 1 following this section. It was also important for us to understand how the various back-end technologies worked with the client side interface and database storage options.

One of the first technologies we explored was Ruby on Rails. However, it was soon discovered that Ruby's age of being the prime Internet technology had already passed. Furthermore, it was no longer properly configured on the WPI CCC servers, which made configuring a standard project for testing and exploration difficult. Our research of these technologies, in conjunction with an understanding that the framework generated within a Ruby on Rails application would be too cumbersome for our project, encouraged us to continue our exploration of other technologies.

The next technology to receive a great deal of experimentation and even system modeling and demo creation from us was Adobe Flex. Having explored Adobe Flex in conjunction with Java Spring framework while interning at Fidelity over the summer, Jason had some experience with the technology and believed that it might be a viable option. Flex provided native grid components and was designed to handle large amounts of data passed in XML. However, it soon became apparent that without the support of a technology team and support staff the ease of use was not equivalent to what had been experienced in the enterprise environment over the summer. Additionally, as Adobe Flex products are compiled into Flash there was some concern as to the cross platform and mobile capability. It was also during this time that we began discussions with WPI CCC server administrators to ensure that our application would have a long term home where it could be hosted and utilized by the WPI Helpdesk. Due to security and maintenance concerns they were unwilling to provide and support a full Java Spring environment. They would however be willing to consider a lighter Java environment if there was a demonstrated business need. These concerns, in addition to our recent introduction to Google Application Engine (GAE) and Google Web Toolkit (GWT), prompted us to make a final development technology transition.

Having recently completed a much smaller Web application in the course Webware (CS-4241) using GAE and GWT, we made a final decision that it was the most appropriate Web technology given the scope and requirements of our application. CCC Administrators were also willing to consider the long term use of GAE on WPI machines. Google Application Engine is Java based and utilizes a local data store that allows the passing and storing of Java objects. It was a relatively new technology but had an active user base which we believed would be helpful in development. Additional details on GAE, GWT, and the additional GUI extension that we selected can be found in the technology implementation section.

Table 1: Technology Comparison

Name	Application	Pros	Cons	Language	Other Notes	WPI Supported
Java Server Pages (JSP)	Web	- Platform independent - Custom tag libraries - ODBC & JDBC	- Virtual Hosting can cause problems - Steep learning curve	Java	Pages are compiled into Java Servlets.	No
ASP.Net	Web	- Mobile Device Support - Server Rendered pages - Login control (2.0)	- Unfamiliar languages - Proprietary	.Net Languages		No
Adobe Flex	Web	- Cross browser standardization (through flash) - Familiarity	- Possible x64 issues	Action Script	Recommended Spring backend	No
Spring (MVC)	Server	- Integrates with JSP, or Flex	- Fairly new technology - Extensive technology to learn in limited time frame	Java		No
JavaServer Faces (JSF)	Server			Java		No
Java Servlet	Server			Java		No
MySQL	Database	- Available - Familiarity		SQL		Yes
Active-Server Pages (ASP)	Web	- ODBC	- Platform Dependant - Memory Usage - Dead…	VBScript		No
Perl Server Pages (PSP)	Web		- Complex syntax	Pearl		No
PHP	Web	- Active community - past experience	- Error Handling, - Guaranteed NO WPI support	PHP		No
Oracle	Database			SQL		Yes
Access 2007	Database		- Backups	SQL/UI		Yes
Groovy	Web			“Javaish”	New (2007)	Maybe
Struts	Server			Java		No
Java Beans	Server			Java		No

Technology Research References

Name	URL(s)
Java Server Pages (JSP)	http://Java.sun.com/developer/technicalArticles/xml/WebAppDev/ http://Java.sun.com/products/jsp/jsp-asp.html http://Java.sun.com/products/jsp/jsp_jservlet.ds.html http://www.hostreview.com/guides/General_Information/articles/081212choosingaWeb.html http://www.scribd.com/doc/2294893/JSP-Presentation
ASP.Net	http://www.startvbdotnet.com/aspsite/asp/features.aspx http://www.w3schools.com/aspnet/aspnet_newfeatures.asp
Adobe Flex	http://www.adobe.com/devnet/flex/
Spring (MVC)	http://www.springsource.org/
JavaServer Faces (JSF)	http://Java.sun.com/Javaee/Javaserverfaces/
Java Servlet	http://Java.sun.com/products/servlet/
MySQL	http://www.mysql.com/
Active-Server (ASP)	http://Java.sun.com/products/jsp/jsp-asp.html http://www.hostreview.com/guides/General_Information/articles/081212choosingaWeb.html
Pearl Server Pages (PSP)	http://www.hostreview.com/guides/General_Information/articles/081212choosingaWeb.html
PHP	http://www.hostreview.com/guides/General_Information/articles/081212choosingaWeb.html
Groovy	http://groovy.codehaus.org/
Java Beans	http://Java.sun.com/developer/technicalArticles/xml/WebAppDev/

3 System Design

3.1 System Functionality

3.1.1 Scenarios

In order to adequately prepare for the development of the application we began by creating a variety of likely scenarios that our system would encounter. We aimed to be as inclusive as possible. The scenarios' titles are listed below and described in full detail in Appendix A: Scenarios.

- User Applies to Helpdesk
- Manager Reviews New Application
- Manager Posts
- Manager Hiring Decision
- User Time Sheet
- Manager Time Sheet
- Manager Performance Review
- User Performance Review

3.1.2 Use Cases

Continuing from the scenarios we created use cases. These were more defined situations that allowed us to fully grasp the flow of the system from a user perspective and allow us to model how we wanted to then perform those actions on the system level, both client side and via the associated server function calls. The use cases we created are listed here and described in detail in Appendix B: Use Cases.

GENERAL:

- Login

USER PROFILE:

- Employee Adds/Updates content in user profile
- Manager Adds/Updates content in user profile

TIMESHEET:

- View Timesheet
- Add Special Case
- Edit Special Case
- Remove Special Case
- Submit Timesheet
- Timesheet Approval
- Print All Timesheets
- Select Timesheets to Print
- Deselect Timesheets from list of selected timesheets
- Print Selected Timesheets
- Printing Status Report
- Mark Printed Timesheets as Printed
- Reprint Timesheets

PERFORMANCE REVIEWS:

- Create Template from Scratch
- Create Template from Existing Template
- Add New Question to Template
- Add Existing Question to Template
- Remove Question from Template
- Finalize Template
- Select Managers to fill out reviews
- Select Employees to Deploy Reviews
- Deploy Review
- Change Visibility on Individual Review
- Change Visibility on All Reviews
- Complete Review
- View All Completed Reviews

HIRING:

- User submits application
- Manager Views Application
- Manager Adds notes to Application
- Manager Adds Ratings to Application
- Manager Accepts Application

3.1.3 Data Model Diagrams

The following diagrams display our initial design for the application and data store objects. They represent the breakdown of the application data into appropriate Java objects that we are able to pass between client and server within the application. It also displays the specific data attributes stored within each of these objects.

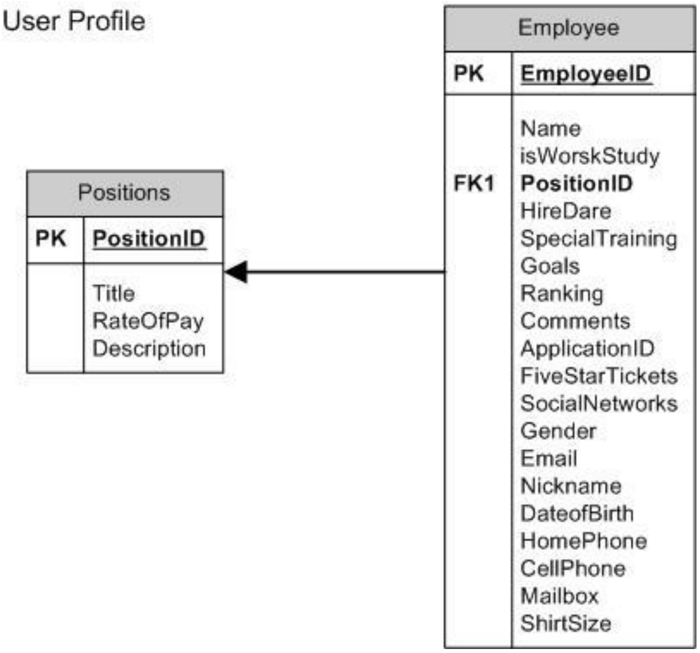


Figure 1: User Profile Data Model Diagram

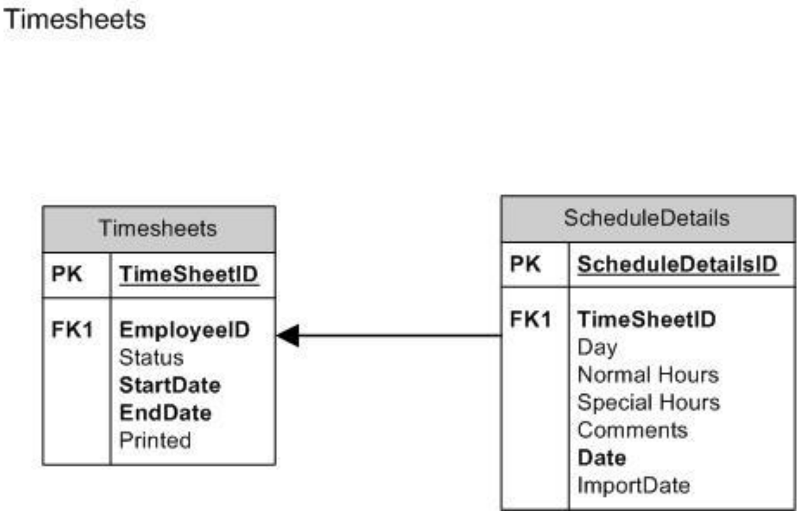


Figure 2: Timesheet Data Model Diagram

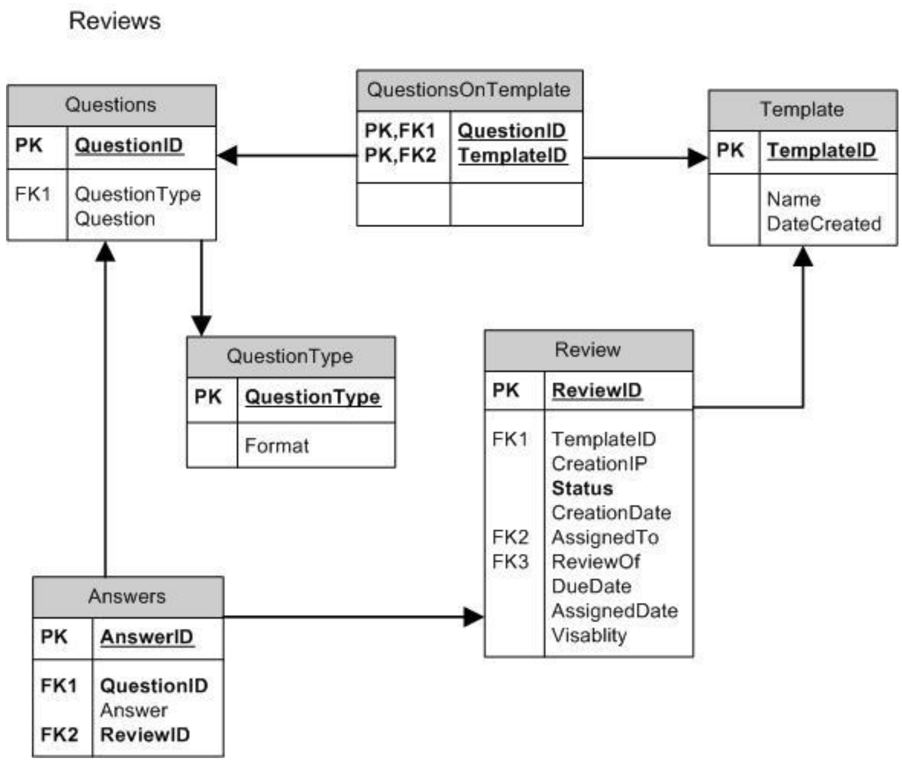


Figure 3: Review Data Model Diagram

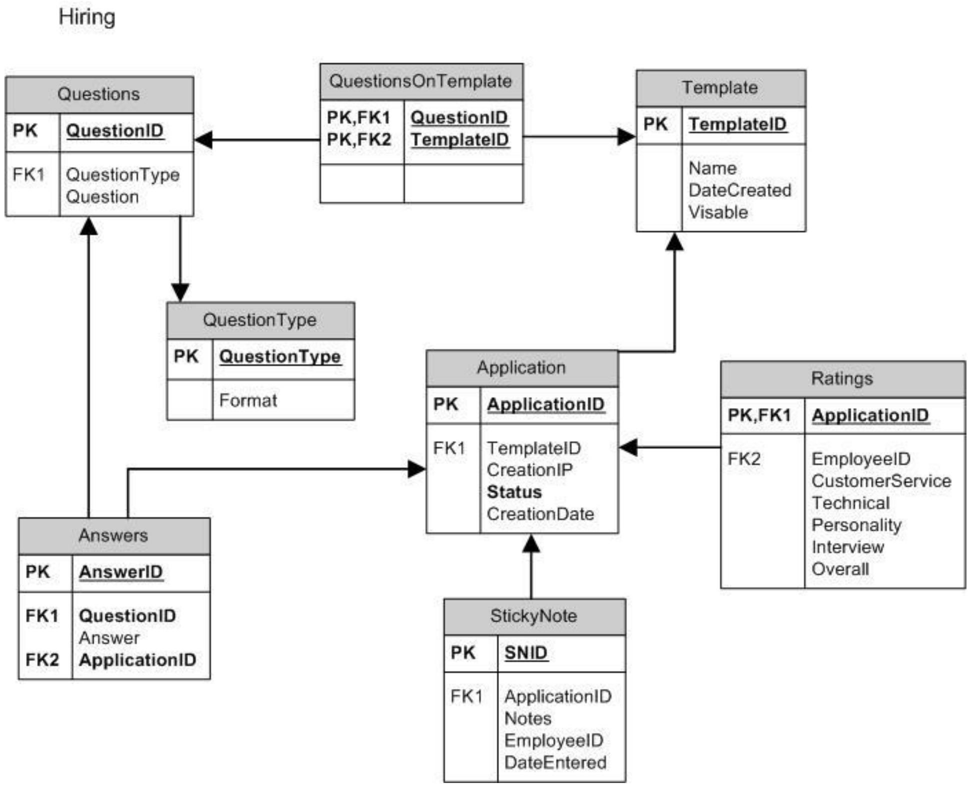


Figure 4: Hiring Data Model Diagram

3.1.4 Prioritization of Features

Before we began development we also felt it was very important to prioritize the features as much as possible. This listing displays the priority levels that were assigned to each of the functionality requirements with discussion and input from WPI Helpdesk management. This prioritization allowed us to focus first on those features thought to be most valuable to our end customer, the WPI Helpdesk. The system components were prioritized in the order they are listed (User Profile, Timesheets, Performance Reviews, and Hiring Process). User Profile was determined to be the highest priority as it contains all employee data and is the primary referenced object for each user. The remaining components were placed in order of importance relayed to us by the WPI Helpdesk Management. Within each component the specific features were prioritized in order of system development necessity and are arranged as such with denotations starting with the letter A.

3.1.4.1 System Functionality:

User Profile:

A Create an user profile from an application

B Allow multiple 5 star tickets

C Allow multiple social networks

Timesheets:

A Export to a universally printable object

B Ability to create blank timesheet

C Pull data from the WhenToWork schedule to generate timesheet

D Generate default time sheet for each employee

E Auto assign new timesheets to employees

F Printing super scripts, and notes with any special cases

G Calculate gross pay

Performance Reviews:

A Create reviews based off of existing templates

B Bulk assign review to employees

Hiring Process:

A Display visible templates

B Notify manger of new applications

C CAPTCHA[4] protection

3.1.4.2 Manager Functionality:

User Profile:

A Ability to manually create user profile

B Add Positions for Helpdesk workers

C Modify Positions for Helpdesk workers

D Has exclusive edit privileges on an Employee's:

- Name
- Gender
- Date of Birth
- Federal Work Study vs. Not
- Hire Date
- Ranking
- Ranking Comments
- Goals
- 5-Star Tickets

Time Sheets:

A Ability to print default/blank timesheets for individuals

B Print select time sheets

C Print all time sheets

D Approval time sheets

E Mark timesheets as printed

F Archive timesheets for a pay period

Performance Reviews:

A Create Templates

B Finalize Template

C Ability to bulk issue user reviews

D View User's reviews

E Complete Outstanding Reviews

F Save In Progress Reviews

G Save Templates

H Modify Templates

I Set visibility on Manager reviews

J View Users who have not Completed their User Review

Hiring Process:

A Create Sticky Notes

[4] CAPTCHA is a robot prevention measure used on Web forms that requires users to enter a word displayed in an image before submitting. More details can be found at http://captcha.net

B Receive Notification of new applications
C Delete an Application
D Automatic Job Offer/Rejection notifications as Helpdesk@wpi.edu

3.1.4.3 Employee Functionality:

User Profile:

A Can set the initial values of:

- Gender
- Date of Birth

B Does not have edit privileges on own:

- Name
- Gender
- Federal Work Study vs. Not
- Hire Date
- Ranking
- Ranking Comments
- Goals

Time Sheets:

A Ability to submit timesheet
B Ability to add/edit/remove special cases

Performance Reviews:

A Complete Outstanding Reviews
B Save In Progress Reviews
C View Review History

3.2 User Interface

Our goal when designing the user interface was to create a Web experience similar in style and response to a desktop application. We created windows and panels in a fashion similar to what would be found a standard non-Web application. This design decision was made to provide for the greatest interactivity and ease of use possible while also making it entirely Web enabled and accessible from anywhere with Web access. We believed this was a critical feature to ensure maximum accessibility to the data and tools available within the system. The system consists of a home login screen utilized by all system users to gain entry into the system. Once logged in, the system displays a tabbed interface that is customized with the appropriate features for either a student employee or manager. The employee tabs provide access to the currently implemented features of the user profile, timesheets, and settings. The manager login includes tabs of user profile listing, timesheets, positions, accounts, and settings. Within each tab is the associated functionality and can include interactive grids described in more detail in the system components. The color scheme chosen was a combination of the WPI colors of grey and crimson with the common application light blue found in most MS Office applications and provided with our EXT GWT GUI development technology.

4 Implementation

4.1 Technology

Our technology exploration allowed us to investigate and review a wide variety of available solutions for implementing a Web application. While reviewing these technologies, and in making the final selection of technology, there were a few different characteristics that were important to consider. First, it was essential to choose a technology that would work well with the functional requirements for the system. It was clear that the system would need to pass data objects from the database to the client, and that the client would often display that data in grids or sorted lists. We also wanted to choose a technology that we had some familiarity with to provide us a fundamental understanding but was also new and would provide us opportunities to learn a different technology. This combination of finding the best fit of the functional and data requirements while simultaneously maximizing the technical learning experience led us into the selection of the technologies we describe below.

4.1.1 Google App Engine

The server side application development was completed using Google App Engine. Google App Engine is a Java development stack developed by Google for the primary purpose of creating AJAX[5] enabled Web applications and is hosted on Google's distributed Web servers, called the cloud. Local development of the Java code is assisted by an Eclipse Development plug-in that allows applications to be deployed directly to the Google App Engine Web servers and made publicly available. This plug-in also has a hosted browser mode so that testing can be done locally without deploying to the cloud. App Engine has a same host origin restriction for data which means data is stored in Google's Big Table database and cannot be stored in remote databases. Due to its ease of integration with the Eclipse development environment and ease of deployment we found it to be the best choice for the server side environment. The Employee Management System was created using the Eclipse plug-in and a local instance of the App

[5] AJAX – Asynchronous JavaScript and XML is a system architecture that supports dynamic client/server communications

Server on our development machines. While data can only be persisted and maintained on the Google data store for deployed instances of the application, Google did provide an admin console for the local server that allowed for review and removal of data via a Web interface. App Engine programs are written in a limited Java so they can be hosted on a Tomcat server. We experimented with deploying our application on a local Tomcat server within the WPI Helpdesk throughout the process.

4.1.2 EXT GWT

The technology that was determined to be the most appropriate and effective for the client side of the Employee Management System was EXT GWT. This is a Java library designed specifically for use in building rich Internet applications in conjunction with the Google Web Toolkit and Google App Engine. It provides a library of highly functional and customizable widgets that are cross browser and platform compatible. The client side code is written in Java and designed using a series of panels and forms which can be attached to one another. Overlapping windows are used as well to provide an interactive desktop like application environment. At compile time the Java is compiled into JavaScript and HTML. EXT GWT is based on the EXT-JS JavaScript library which provides JavaScript functionalities and enables interactivity through grids with sortable columns, drag and drop listings, information pop-ups and resizable windows. EXT GWT also allows asynchronous calls to be made to the server and is used for the transfer of Java data objects containing such information as login verification, user profile details, and timesheet information. Finally, a series of non-obtrusive popup notifications are utilized to inform the user with vital information such as success of a change, or any failures that are non-critical to functionality.

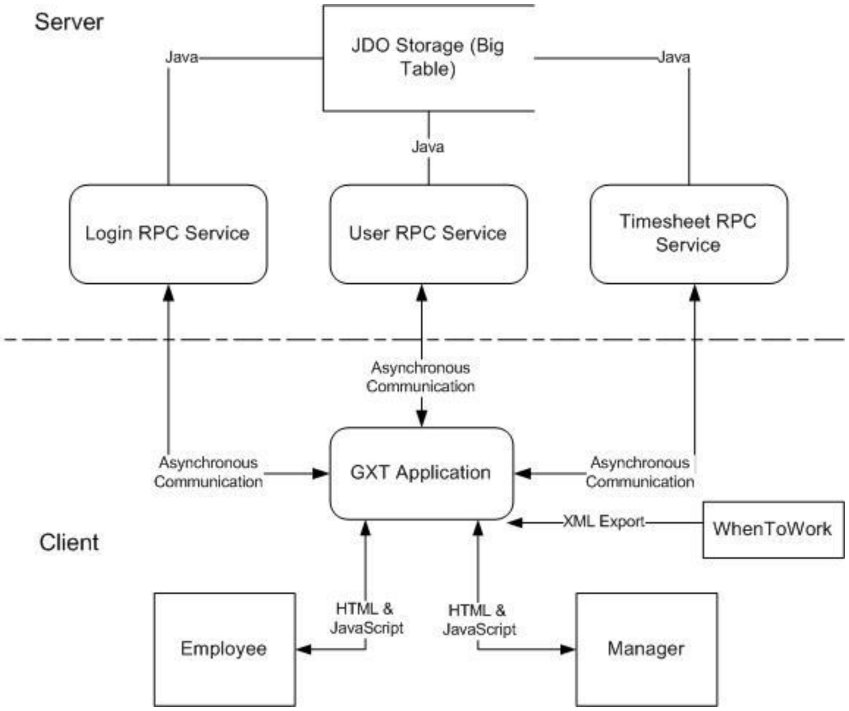
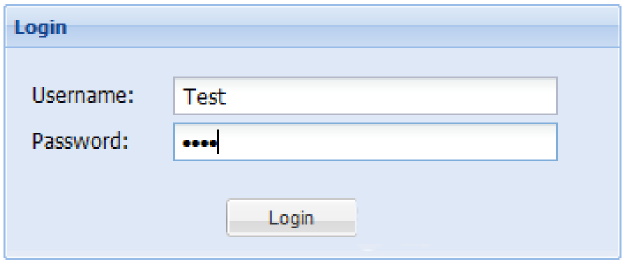


Figure 5: System Architecture

4.2 System Components

This section provides an in depth explanation of the system components that were implemented for the MQP. The primary components that were completed in full development were the account management, employee profile, and timesheet management. Each system is a combination of the server-side servers working in conjunction with the browser-based client application.



Login

Username: Test

Password: ••••

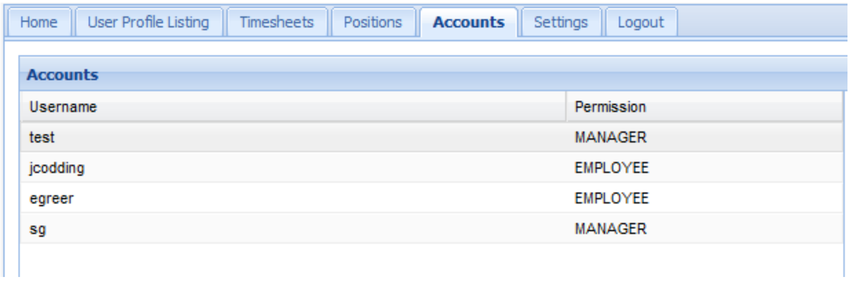
Figure 6: Login Form

4.2.1 Login/User Accounts

One of the first and most important components developed was the user account and login system. In order to gain access to the system a set of username and password credentials must be provided by the user to the main login screen, Figure 6: Login Form, which is displayed upon first view of the Web application. Once these credentials are provided the client asynchronously passes the username and password that were provided to the server. This is done with SHA512 encryption to ensure security of the user's password. Once on the server the validation is then completed by comparing the stored encrypted password with the password that was provided. If incorrect, a failed validation message will be sent back to the client, which is displayed as a system notification popup. If the validation passes, however, an account object with a permissions flag is sent back to the client. This instructs the client to continue with logging into the system and either display the employee tabs and features if a standard employee or the manager tabs and features if a manager. These permissions are then stored locally so that they can be verified as the user navigates within the system.

4.2.1.1 *Account Management*

The account management component of the system, Figure 7, is used by the manager to view, modify, and create system login accounts with the desired permissions. It provides a listing of all employees as well as their permissions and the ability to reset their passwords. From here the manager can also create new user accounts which are associated with the corresponding user profile. This is completed via a wizard-style interface to ensure maximum ease of use for the manager.



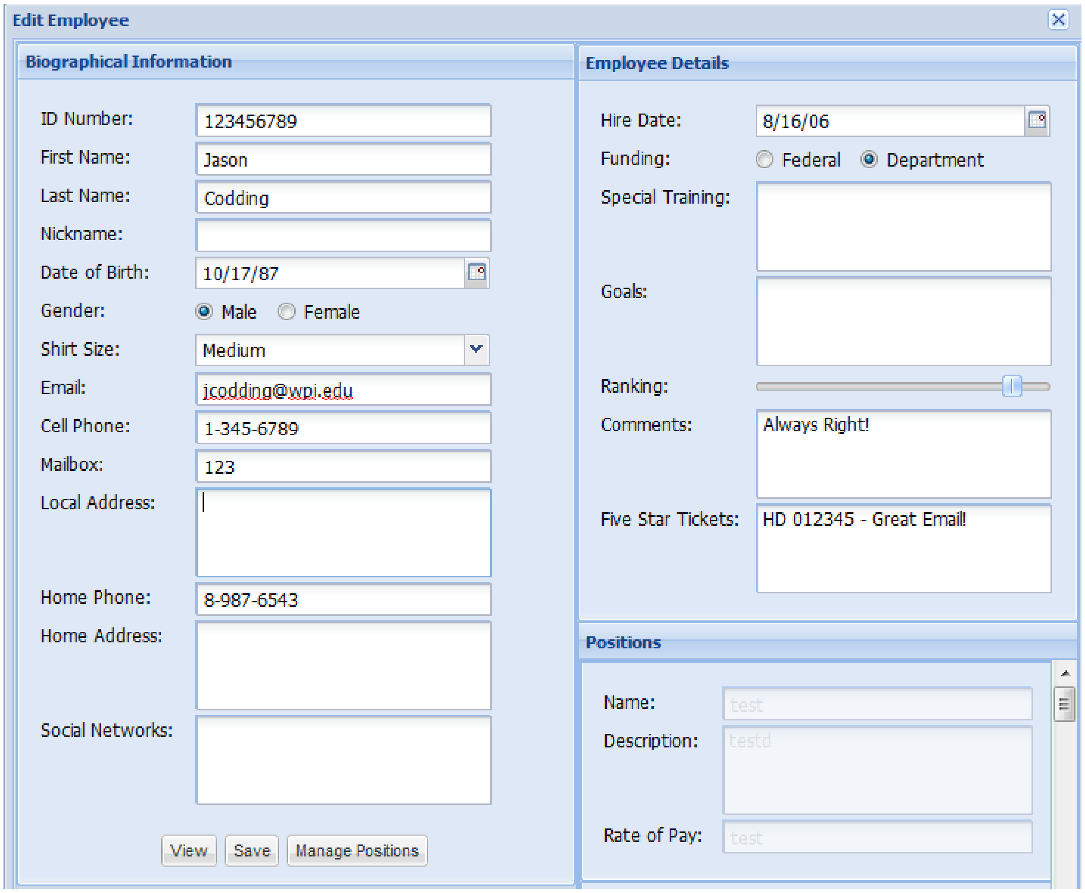
Accounts

Username	Permission
test	MANAGER
jcodding	EMPLOYEE
egreer	EMPLOYEE
sg	MANAGER

Figure 7: Account Listing

4.2.2 User Profile

The user profile for each employee is accessible by both the associated user and the manager. It is the central location where all of the details about a user can be retrieved. This form can be seen in Figure 8. It contains three sections: the biographical information, the employee details, and their positions. The biographical information section includes: first name, last name, nickname, date of birth, gender, shirt size, email, cell phone, mailbox, local address, home phone, home address, and social network details. The employee details section includes: hire date, funding source, special training, goals, ranking, comments, and five star ticket listing. The positions box displays the current positions that were associated with the user by the manager.



Edit Employee

Biographical Information

ID Number: 123456789
First Name: Jason
Last Name: Codding
Nickname:
Date of Birth: 10/17/87
Gender: ● Male ○ Female
Shirt Size: Medium
Email: jcodding@wpi.edu
Cell Phone: 1-345-6789
Mailbox: 123
Local Address:
Home Phone: 8-987-6543
Home Address:
Social Networks:

Employee Details

Hire Date: 8/16/06
Funding: ○ Federal ● Department
Special Training:
Goals:
Ranking:
Comments: Always Right!
Five Star Tickets: HD 012345 - Great Email!

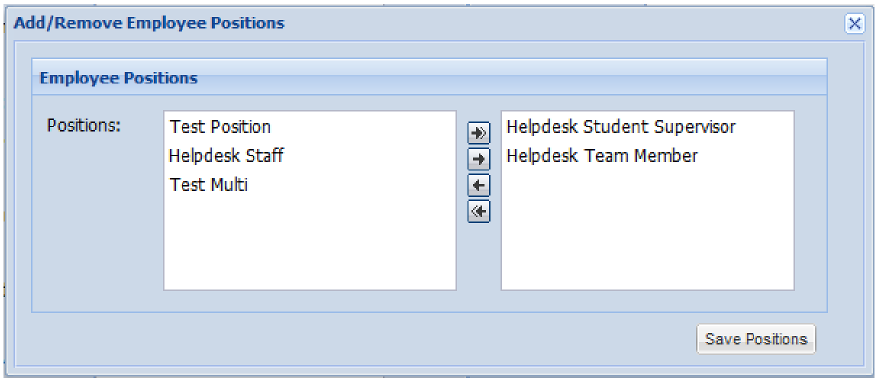
Positions

Name: test
Description: testd
Rate of Pay: test

Figure 8: Employee Profile From

4.2.2.1 Employee

While logged into the system as an employee you are able to view the user profile for your account. You can view all fields in the profile. However, editing is limited to the nickname, shirt size, email, cell phone, mailbox, local address, home phone, home address, and social networks. This allows employees to edit that information which might change while employed but ensures the integrity of employment data.



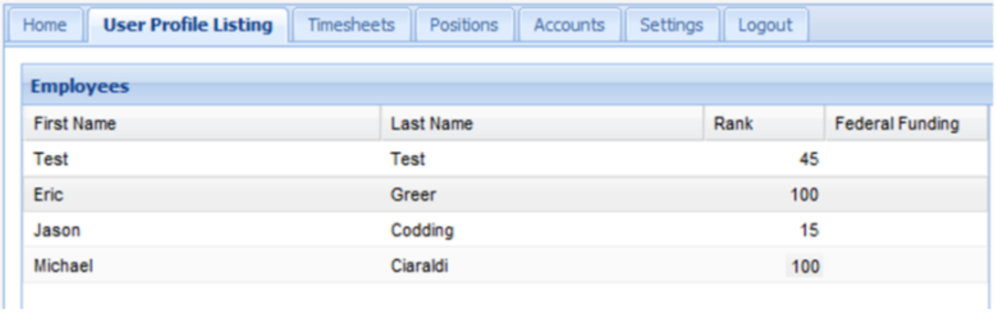
Test Position
Helpdesk Staff
Test Multi

Helpdesk Student Supervisor
Helpdesk Team Member

Figure 9: Position Management

4.2.2.2 Manager

The manager has the ability to view the same user profile screen for each of the employees in the system. The manager has increased edit capabilities so that he can correct incorrect data and record the details of the employment in the employee details section where they can be viewed by the user. The manager also has the ability to add or delete positions associated with any particular user. This is done through a drag and drop style interface, Figure 9, made available to him in a window that can be launched from this screen. The manager sees a listing of all employees, Figure 10 , in a sortable grid and can double click on a particular employee to launch their profile to view the details and perform any changes.



Employees

First Name	Last Name	Rank	Federal Funding
Test	Test	45	
Eric	Greer	100	
Jason	Codding	15	
Michael	Ciaraldi	100	

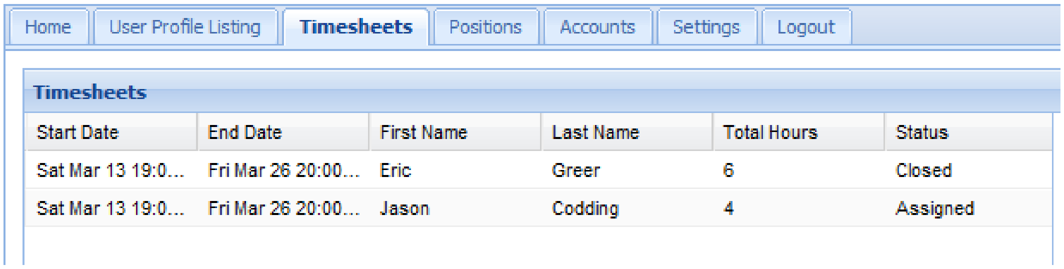
Figure 10: Employee Profile Listing

4.2.3 Timesheets

The timesheet section of our application provides a semi-automated Web based solution to what was previously completed via paper and manual calculations. The timesheet process begins with the import of the shifts and hours data from the WhenToWork scheduling system. The XML is then parsed by the server and the data is stored appropriately as timesheet detail objects for each user. The timesheet consists of a set of schedule details for each day of the week in that timesheet period. Each schedule detail may then have an exception stored with it that is entered by either the user or manager and signifies a change from the standard working shift as imported. The timesheet itself knows about the date range, associated employee, and its status which can be set to assigned, approved, or closed.

4.2.3.1 Employee

When navigated to the timesheet tab the employee may view his/her own timesheet and see its status. This is available for both the current and past timesheets. The hours worked are automatically completed by the system utilizing the details imported from the When2Work scheduling system. They can be edited by the user and a comment clarifying the reason for the change can be made by the employee for later review by the manager. Once the user believes the times recorded are accurate he or she may select the ‘submit’ button pushing the timesheet into the manager’s queue.

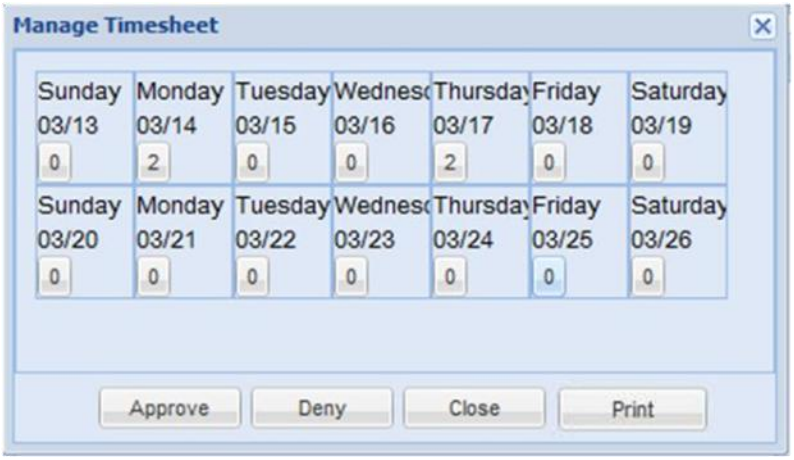


Start Date	End Date	First Name	Last Name	Total Hours	Status
Sat Mar 13 19:0...	Fri Mar 26 20:00...	Eric	Greer	6	Closed
Sat Mar 13 19:0...	Fri Mar 26 20:00...	Jason	Codding	4	Assigned

Figure 11: Manager Timesheet Listing

4.2.3.2 Manager

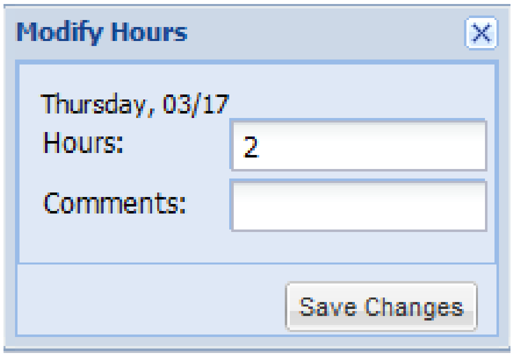
The manager is able to view all timesheets for employees, regardless of the current status Figure 11. There are two primary listings made available to him in the timesheet tab: the current timesheets and an archived listing. The manager is able to retrieve timesheets from this listing where he can view all of the recorded hours, and the details for that timesheet Figure 12.



Sunday	Monday	Tuesday	Wednesc	Thursday	Friday	Saturday
03/13	03/14	03/15	03/16	03/17	03/18	03/19
0	2	0	0	2	0	0
Sunday	Monday	Tuesday	Wednesc	Thursday	Friday	Saturday
03/20	03/21	03/22	03/23	03/24	03/25	03/26
0	0	0	0	0	0	0

Figure 12: Timesheet View

Any exceptions to the normal hours are highlighted to bring attention to them and the manager can view any comments left by the employee, Figure 13, explaining the reason for the change. The manager can also modify the hours as needed. Once satisfied with the timesheet it can be marked as approved. If the manager is not satisfied with what was submitted it may be alternatively returned to the user by 'denying' the submission and returning it to assigned status. After reviewing the timesheets the manager is able to print the timesheets to be signed via a PDF document in the standard format accepted by WPI Accounting, and example of this printing is found in Appendix C: Screenshots, Figure 14. This can be done one at a time or as a complete set for all workers.



Thursday, 03/17
Hours: 2
Comments:

Figure 13: Timesheet Details

5 Future Development

5.1 Deployment Instructions

Code Development:

1. Download Eclipse
2. Install the Google App Engine Plug-in
3. Import the project from the Zip file as a GWT Project
4. Run the application as a "Web Application" to view the project in hosted mode.

Deploy Eclipse Project to App Engine

1. Click the "Deploy to App Engine" button
2. "Click on App Engine project settings…"
3. Specify the Application ID and version number
4. Enter your Email address and Password
5. Click Deploy

Deploy Project to Tomcat Server

1. Click the "GWT Compile Project" button
2. Select the Project to Compile and click "Compile"
3. After the compilation completes copy the war folder contents to the Tomcat Server

5.2 Know Issues

The existing system contains a few known issues that were unable to be solved in the available development time frame. They should be the first priority for future development efforts. These issues are as follows:

5.2.1.1 User Profile:

- When adding or modifying an employees' position as a manager the positions assigned to the user do not actively refresh on the profile window after modifications are made in the pop-up. However, the modifications are persisted to the server and data store.

- Slider for user rank on user profile does not slide.

- Validation feedback needs to be done on the user profile form before submitting something with error. Currently no feedback is provided, nor data persisted, if invalid data is submitted.

- User profile listing doesn't show work study status

5.2.1.2 Timesheet:

- Can't upload files; XML data must currently be copied and pasted into a textbox for importation.

- On XML Import if multiple shifts exist on one day they are overwritten, not added together

- When hours are different from the imported WhenToWork hours the manager's view of the timesheet doesn't highlight this change with a distinct color as expected.

5.3 Additional Functionality

Future system development includes implementation of the hiring process and review management. These functions are designed and described throughout this document and stub classes have been created for the functionality in the code base. In addition, prior to utilization of the system at the WPI Helpdesk it will need to be ported over to a local server. This is to ensure the security of the data contained in the system such as personal and contact information on each employee. Testing was successfully completed to validate that the application can be run on a local Apache Tomcat server. Operating on a local server will allow for storage to a WPI MySQL database.

Appendix A: Scenarios

User Applies to Helpdesk:

1. User enters application information online
2. System: Stores information to DB, and notifies Manager

Manager Reviews New Application:

1. Receives notification about new application
2. Log onto the System
3. *Navigates to applications tab*
4. Selects a pending application
    a. Accepts Application (Not Spam, Has Basic Requirements)
    b. Denies Application
5. Schedules interview/sends rejection letter

Manager Posts Comments:

1. Log onto the System
2. *Navigates to applications tab*
3. Select the application
4. Selects add Comments
5. Adds comments to the dialog box
6. Saves Comments
7. Clicks on Edit Ratings
8. Adds Ratings (Quantitative Data)
    a. Customer Service
    b. Technical
    c. Personality and Fit rating
    d. Interview Rating
    e. Overall Score
9. Saves Ratings

Manager Hiring Decision:

1. Log onto the System
2. *Navigates to applications tab*
3. Select the application
    a. Makes hiring offer
    b. Declines to make offer
4. Prompt to edit default message
5. Send Appropriate message
6. Saves Application

User Time Sheet

1. Log onto System
2. *Navigate to Time Sheet Tab*
3. Add special cases if Any
    - *a.* Enter Comments
    - *b.* Enter New number hours, for correct day
    - *c.* Save special cases
    - *d.* Repeat as needed
4. Submit Timesheet for approval

Manager Time sheet:

1. Log onto System
2. *Navigate to Time Sheet Tab*
3. Look at time sheets needing approval
    - *a.* Approve timesheet
4. Look at list of Time Sheets to be printed
5. Check to see if any are missing
    - *a.* If one is missing, Manager can submit the users default schedule.
6. Print time Sheets
    - *a.* Status report of number printed, and other calculations

Manager Performance Review:

1. Log onto System
2. *Navigate to Performance Review Tab*
3. Create New Review
    - a. Add new question
    - b. Specify Question type
    - c. Enter Question text
    - d. Repeat as necessary
4. Deploy review
    - a. Selects manages to fill out review
    - b. Creates a Manager Review for each profile, Sets initial visibility
    - c. Creates a Self Review for each profile
5. View Reviews
    - a. Sees Uncompleted Manager reviews for all users
    - b. Sees Completed Manager reviews for all users, of the selected Review
    - c. Sees Uncompleted User reviews for all users
    - d. Sees Completed User reviews for all users, of the selected Review
6. Clicks on a Review to complete or view
7. Saves review

8. Changes all Manager Review Visibilities.

User Performance Review:

1. Log onto System
2. *Navigate to Performance Review Tab*
3. View current performance score
4. View Outstanding reviews
5. Click on a performance review to complete.
6. Complete review
7. Submit Review

Appendix B: Use Cases

GENERAL:

Name: Login
Participating Actors: Employee, Manager
Entry Condition: Employee/Manager enters login credentials
Exit Condition: System displays home screen
Flow of Events:

1. Employee/Manager enters login credentials into username/password field
2. System authenticates user against stored credentials.
3. System checks whether given user exists in Helpdesk Database
4. System displays appropriate home screen

USER PROFILE:

Name: Employee Adds/Updates content in user profile
Participating Actors: Employee
Entry Condition: Employee clicks on the User Profile tab
Exit Condition: System displays updated user profile
Flow of Events:

1. Employee clicks on the user profile tab
2. System displays existing user profile data
3. Employee clicks 'Edit' button
4. System displays user profile with editable fields
5. Employee makes all desired edits and clicks save
6. System displays updated user profile data

Name: Manager Adds/Updates content in user profile
Participating Actors: Manager
Entry Condition: Manager clicks on the User Profiles tab
Exit Condition: System displays updated user profile
Flow of Events:

1. Manager clicks on the user profiles tab
2. System displays sorted listing of all users
3. Manager clicks on an individual user from the list
4. System displays existing user profile data
5. Manager clicks 'Edit' button
6. System displays user profile with editable fields
7. Manager makes all desired edits and clicks save
8. System displays updated user profile data

TIMESHEET:

Name: View Timesheet
Participating Actors: Employee, Manager
Entry Condition: Employee/Manager clicks on a timesheet
Exit Condition: Timesheet is displayed
Flow of Events:

1. Employee/Manager clicks on a timesheet
2. System displays selected timesheet

Name: Add Special Case
Participating Actors: Employee
Entry Condition: Employee click on Add Special Case
Exit Condition: Special Case is Saved
Flow of Events:

1. Employee clicks on add special case, and then clicks on the date they want for the case.
2. System displays entry fields for total number of hours, and comments
3. User enters the total number of hours for that day and clicks save
4. System saves modification to the timesheet
5. System displays the timesheet with the change annotated with the comment.

Name: Edit Special Case
Participating Actors: Employee
Entry Condition: Employee clicks on a date on the timesheet
Exit Condition: Special Case is saved
Flow of Events:

1. User clicks on a date on the timesheet
2. System Highlights Date
3. User Clicks Edit
4. System displays entry fields filled with current values for total number of hours and comment
5. User modifies values and clicks save
6. System saves modification to the timesheet
7. System displays the timesheet with the change annotated with the comment.

Name: Remove Special Case
Participating Actors: Employee
Entry Condition: Employee clicks on a date on the timesheet
Exit Condition: Timesheet is redisplayed
Flow of Events:

1. User clicks on a date on the timesheet
2. System Highlights Date
3. User Clicks remove

4. System removes the special case from the timesheet
5. System displays the timesheet with original values.

Name: Submit Timesheet
Participating Actors: Employee
Entry Condition: Employee click on current timesheet
Exit Condition: Timesheet is submitted
Flow of Events:

1. User clicks on the submit timesheet button
2. System changes the status of the timesheet to "Pending"

Name: Timesheet Approval
Participating Actors: Manager
Entry Condition: Manager clicks approve button on a timesheet
Exit Condition: Timesheet is approved
Flow of Events:

1. Manager clicks on approve button
2. System changes timesheet status to "Approved"

Name: Print All Timesheets
Participating Actors: Manager
Entry Condition: Manager Clicks Print All Timesheets
Exit Condition: ***Printing Status Report*** is displayed
Flow of Events:

1. Manager clicks on "Print All Timesheets" button
2. System displays the Window's printer menu
3. Manager selects the printer and clicks print
4. System sends all timesheets to the selected printer

Name: Select Timesheets to Print
Participating Actors: Manager
Entry Condition: Manager clicks on a timesheet in the "Approved queue"
Exit Condition: "Print Selected Timesheets" button is clicked
Flow of Events:

1. Manager clicks on a timesheet
2. System highlights the timesheet
3. Manager Repeat steps 1 (and system repeats step 2) until desired timesheets are selected.
4. Manager clicks "print selected timesheets" button

Name: Deselect Timesheets from list of selected timesheets
Participating Actors: Manager
Entry Condition: Manager clicks on a timesheet in the "Approved queue"
Exit Condition: "Print Selected Timesheets" button is clicked
Flow of Events:

1. Manager clicks on a highlighted timesheet
2. System un-highlights the timesheet
3. Manager Repeat steps 1 (and system repeats step 2) until desired timesheets are un-selected.
4. Manager clicks "print selected timesheets" button

Name: Print Selected Timesheets
Participating Actors: Manager
Entry Condition: Manager Clicks "print selected timesheets" button
Exit Condition: ***Printing Status Report*** is displayed
Flow of Events:

1. Manager clicks on "Print All Timesheets" button
2. System displays the Window's printer menu
3. Manager selects the printer and clicks print
4. System sends selected (highlighted) timesheets to the selected printer
5. System displays ***Printing Status Report***

Name: Printing Status Report
Participating Actors: Manager
Entry Condition: Documents were sent to the printer
Exit Condition: Manager clicks ok, or retry button, or cancelled button
Flow of Events:

1. System displays print job characteristics (Number of sheets being printed….)
2. Manager clicks "ok" If everything printed successfully
3. Manager clicks "retry" if job did not print successfully
4. Manager clicks "cancelled" if job did not print successfully and he doesn't want to retry, or if he doesn't want to mark the sheets as printed.

Name: Mark Printed Timesheets as Printed
Participating Actors: Manager
Entry Condition: Manager clicks "ok" on the Printing Status Report
Exit Condition: System Reorders timesheets in the "Approved Queue"
Flow of Events:

1. Manager Clicks the OK button on the Printing Status Report
2. System marks all time sheets that were sent to the printer as printed.
3. System reorders the timesheets in the "Approved Queue" so that the printed ones are on bottom.

Name: Reprint Timesheets

Participating Actors: Manager

Entry Condition: Manager clicks on "retry" on the Printing Status Report

Exit Condition: ***Printing Status Report*** is displayed

Flow of Events:

1. Manager clicks on "retry" on the Printing Status Report
2. System Displays windows Printer menu
3. Manager selects the printer
4. System sends the documents to the printer.

PERFORMANCE REVIEWS:

Name: Create Template from Scratch

Participating Actors: Manager

Entry Condition: Manager Clicks on Create Template

Exit Condition: Template is created

Flow of Events:

1. Manager Clicks on create template
2. System Prompts to copy existing template, or new template
3. Manager selects new template, and clicks start
4. System displays the Modify Template Screen with no questions.

Name: Create Template from Existing Template

Participating Actors: Manager

Entry Condition: Manager Clicks on Create Template

Exit Condition: Template is created

Flow of Events:

1. Manager Clicks on create template
2. System Prompts to copy existing template, or new template
3. Manager selects template to copy, and clicks start
4. System copies the selected templates question
5. System displays the Modify Template Screen with the existing questions.

Name: Add New Question to Template

Participating Actors: Manager

Entry Condition: Manager Clicks on Add Question

Exit Condition: Question is added to the template

Flow of Events:

1. Manager clicks on Add Question
2. System prompts to select existing question, or create new question
3. Manager selects create new question

4. System displays entry boxes for question fields
5. Manager fills in fields and clicks add.
6. System creates new question and associates with template
7. System returns to the Modify Template Screen.

Name: Add Existing Question to Template
Participating Actors: Manager
Entry Condition: Manager Clicks on Add Question
Exit Condition: Question is added to the template
Flow of Events:

1. Manager clicks on Add Question
2. System prompts to select existing question, or create new question
3. Manager selects a question from list of questions, and clicks Add
4. System associates question with template and returns to the Modify Template Screen.

Name: Remove Question from Template
Participating Actors: Manager
Entry Condition: Manager Clicks on Remove Button
Exit Condition: Question is removed from template
Flow of Events:

1. Manager clicks on remove button next to question
2. System removes the question from template, and removes it from the Modify Template Screen.

Name: Finalize Template
Participating Actors: Manager
Entry Condition: Manager clicks on the finalize template button
Exit Condition: System brings manager to deploy review screen
Flow of Events:

1. Manager clicks on the finalize template button
2. System prompts for confirmation
3. Manager confirms finalization
4. System finalizes marks the template as finalized, System displays deploy review screen

Name: Select Managers to fill out reviews
Participating Actors: Manager
Entry Condition: Manager clicks on “Select Managers”
Exit Condition: Return to deploy review screen
Flow of Events:

1. Manager clicks on Select Managers
2. System displays list of managers
3. Manager selects which managers need to fill out review, and clicks save
4. System saves the managers to fill out the review and returns to the deploy review screen

Name: Select Employees to Deploy Reviews

Participating Actors: Manager

Entry Condition: Manager clicks on “Select Employees” button

Exit Condition: Return to deploy review screen

Flow of Events:

1. Manager clicks on select employees
2. System displays employee selection menu
3. Manager selects employees and then clicks save
4. System saves employees and returns to the deploy review screen

Name: Deploy Review

Participating Actors: Manager

Entry Condition: Manager Clicks deploy review

Exit Condition: System displays confirmation message

Flow of Events:

1. Manager clicks deploy review
2. System displays summary of the deployment
3. Manager clicks confirm
4. System creates reviews and assigns them as designated, and then displays confirmation message.

Name: Change Visibility on Individual Review

Participating Actors: Manager

Entry Condition: Manager Click s on a review assigned to them.

Exit Condition: System changes the visibility for the review

Flow of Events:

1. Manager click on a review assigned to them
2. System displays review
3. Manager clicks on the Share button, and saves the review
4. System sets the visibility to Shared.

Name: Change Visibility on All Reviews

Participating Actors: Manager

Entry Condition: Manager clicks on the Share all Reviews button

Exit Condition: System changes the visibility for the review

Flow of Events:

1. Manager clicks on the Share all Reviews button
2. System sets all visibilities for review of the current template to Shared.

Name: Complete Review
Participating Actors: Manager/Employee
Entry Condition: Manager/Employee clicks on an Uncompleted Review
Exit Condition: Manager/Employee Submits Review
Flow of Events:

1. Manager/Employee clicks on an Uncompleted Review
2. System displays review
3. Manager/Employee fills out questions, and clicks submit
4. System records all information, and marks the review as complete

Name: View All Completed Reviews
Participating Actors: Employee
Entry Condition: Employee clicks on "Review History"
Exit Condition: System displays reviews
Flow of Events:

1. Employee clicks on "Review History"
2. System displays all reviews that have been completed.

HIRING:

Name: User submits application
Participating Actors: Applicant
Entry Condition: Applicant navigates to application Webpage
Exit Condition: Systems displays notification of successful submission
Flow of Events:

1. Applicant navigates to application Webpage
2. System displays application
3. Applicant completes application and hits submit
4. System stores application and displays notification of success to applicant

Name: Manager Views Application
Participating Actors: Manager
Entry Condition: Manager clicks Hiring tab
Exit Condition: System displays application
Flow of Events:

1. Manager clicks Hiring tab
2. System displays lists of pending applications and their status
3. Manager selects desired application
4. System displays desired application

Name: Manager Adds notes to Application
Participating Actors: Manager
Entry Condition: Manager clicks Hiring tab

Exit Condition: System confirms update of notes
Flow of Events:

1. Manager clicks Hiring tab
2. System displays lists of pending applications and their status
3. Manager selects desired application
4. System displays desired application
5. Manager clicks add notes
6. System displays notes dialog box
7. Manager enters notes and clicks save
8. System saves new notes and displays confirmation

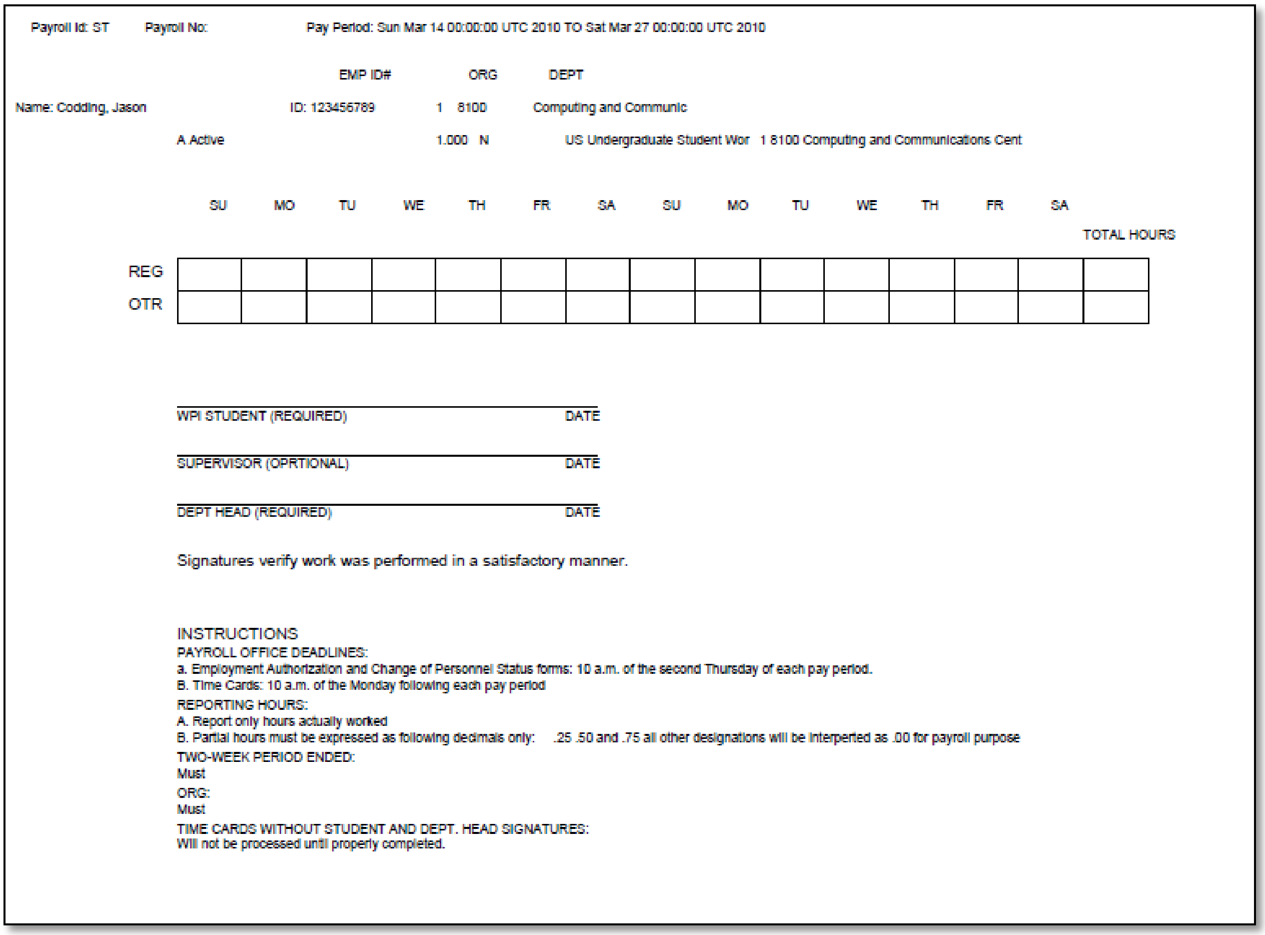
Name: Manager Adds Ratings to Application
Participating Actors: Manager
Entry Condition: Manager clicks Hiring tab
Exit Condition: System confirms update of rating
Flow of Events:

1. Manager clicks Hiring tab
2. System displays lists of pending applications and their status
3. Manager selects desired application
4. System displays desired application
5. Manager clicks edit ratings
6. System displays rating dialog box
7. Manager enters ratings and clicks save
8. System saves new ratings and displays confirmation

Name: Manager Accepts Application
Participating Actors: Manager
Entry Condition: Manager clicks Hiring tab
Exit Condition: System displays application acceptance confirmation
Flow of Events:

1. Manager clicks Hiring tab
2. System displays lists of pending applications and their status
3. Manager selects desired application
4. System displays desired application
5. Manager clicks ‘accept application’
6. System marks application status as accepted and creates user profile
7. System displays confirmation to manager of accepted status

Appendix C: Screenshots



Payroll Id: ST Payroll No: Pay Period: Sun Mar 14 00:00:00 UTC 2010 TO Sat Mar 27 00:00:00 UTC 2010

EMP ID# ORG DEPT

Name: Codding, Jason ID: 123456789 1 8100 Computing and Communic

A Active 1.000 N US Undergraduate Student Wor 1 8100 Computing and Communications Cent

	SU	MO	TU	WE	TH	FR	SA	SU	MO	TU	WE	TH	FR	SA	TOTAL HOURS
REG															
OTR															

WPI STUDENT (REQUIRED) DATE

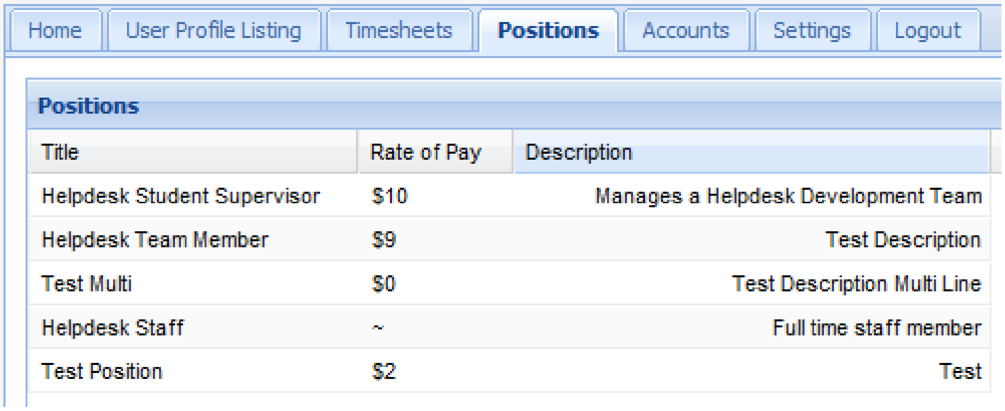
SUPERVISOR (OPRTIONAL) DATE

DEPT HEAD (REQUIRED) DATE

Signatures verify work was performed in a satisfactory manner.

INSTRUCTIONS
PAYROLL OFFICE DEADLINES:
a. Employment Authorization and Change of Personnel Status forms: 10 a.m. of the second Thursday of each pay period.
B. Time Cards: 10 a.m. of the Monday following each pay period
REPORTING HOURS:
A. Report only hours actually worked
B. Partial hours must be expressed as following decimals only: .25 .50 and .75 all other designations will be interperted as .00 for payroll purpose
TWO-WEEK PERIOD ENDED:
Must
ORG:
Must
TIME CARDS WITHOUT STUDENT AND DEPT. HEAD SIGNATURES:
Will not be processed until properly completed.

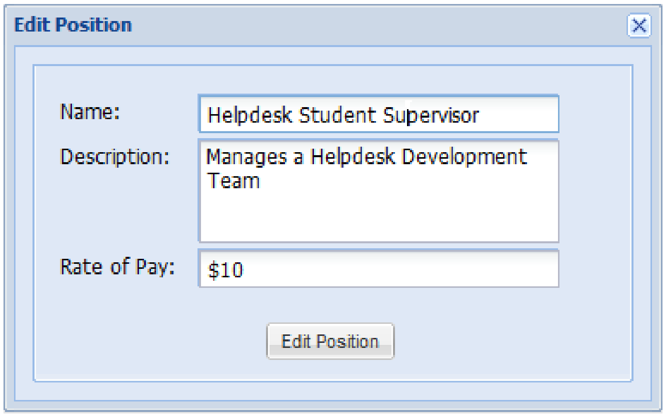
Figure 14: Example PDF Timesheet



Positions

Title	Rate of Pay	Description
Helpdesk Student Supervisor	$10	Manages a Helpdesk Development Team
Helpdesk Team Member	$9	Test Description
Test Multi	$0	Test Description Multi Line
Helpdesk Staff	~	Full time staff member
Test Position	$2	Test

Figure 15: Position Listing



Name: Helpdesk Student Supervisor

Description: Manages a Helpdesk Development Team

Rate of Pay: $10

Figure 16: Edit Position Form

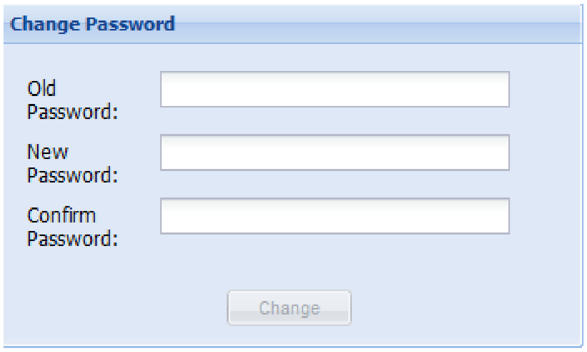


Figure 17: Change Password Form

Appendix D: Timeline

Table 2: Project Timeline

Weeks	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32
Requirements Gathering	🟥	🟥														
System Design		🟥	🟥													
Data Modeling			🟥	🟥												
Technology Research					🟥	🟥										
Technology Evaluation						🟥	🟥	🟥								
Framework Design								🟥	🟥							
Accounts										🟥			🟥			
User Profile											🟥	🟥	🟥			
User Profile Testing												🟥	🟥			
Timesheet													🟥	🟥	🟥	
Time Sheet Testing															🟥	🟥
Report					🟥	🟥							🟥	🟥	🟥	🟥

Appendix E: Detailed Data Requirements

[1] Items are editable by the user

[2] Visible to all other users of the system

User Profile:

- Biographical Information
  - Name [2]
  - Username[1]
  - Nickname [1, 2]
  - Date of Birth
  - Telephone [1,2]
  - Email address[1,2]
  - Gender
  - Local Address [1,2]
  - Home Address [1]
  - Mailbox [1,2]
  - Shirt Size[1]
- Social networking information (AIM, Facebook, LinkedIn…)[1,2]
  - Social Network Name
  - URL/Username
- Position/Team membership
  - Position Name
  - Rate of pay
  - Description
- Employment Details
  - Federal Work Study vs. Not
  - Hire Date
  - Special Training [1?]
  - Ranking
  - Ranking Comments
  - Goals
  - 5-Star Tickets

Time Sheets:

- Schedule
  - Week
  - Day
  - Employee
  - Number Hours
- Status
  - Assigned
  - Pending
  - Approved
  - Printed
  - Archive
- Special cases
  - Week
  - Day
  - Number of hours
  - Comments

Performance Reviews:

- Questions
- Answers
- Templates
    - Name
    - Date Created
- Employee Assigned to
- Employee Review of
- Assigned Date
- Due Date
- Visibility
- Completed

Hiring Process:

- Questions
- Answers
- Templates
    - Name
    - Date Created
    - Visibility
- Creation Date
- Creation IP
- Status
    - Accepted
    - Denied
    - Hired
- Sticky Notes
    - Creating Employee
    - Note
    - Date Created
- Ratings
    - Creating Employee
    - Customer Service
    - Technical
    - Personality/Fit
    - Interview
    - Overall