**CSC 131: Section 1**

**Fall 2014**

**Deliverable # 2**

**AESA Solutions**

**10/16/2014**

**Software Requirement Specification (SRS)**

**1. INTRODUCTION**

**1.1 Purpose of this Document**

This software requirement specification describes the functions and performance requirements allocated to the *White Wave Business Suite*. The *White Wave Business Suite* is a stand-alone system that will provide functionality *White Wave Digital* will need for their daily business practices. Primary functions will consist of a ticket tracking component, a database reference component, and financial statement generation component.

**1.2 Scope of the Development Project**

The *White Wave Business Suite* is a system that will encompass a ticketing system, a database to hold required information like client locations, and some sensitive information like passwords for clients. This program will also have some minor financial functions such as making and sending invoices. This program will be accessible through the internet and possibly mobile devices to aid tech support agents who need information while in the field.

**2. GENERAL DESCRIPTION**

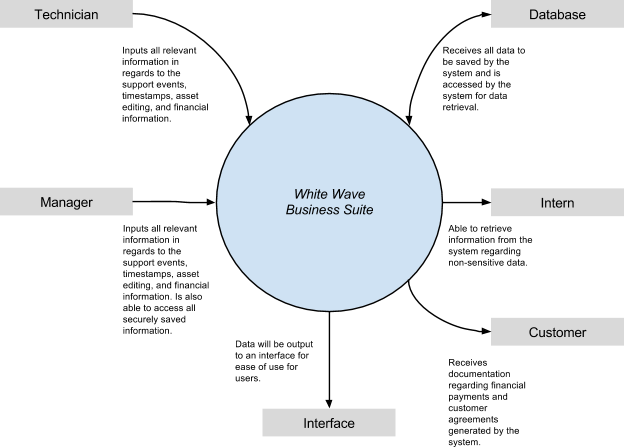
**2.1 Glossary (Definitions, Acronyms, and Abbreviations)**

1. Asset: Internal or external property which can be physical or virtual.
2. Customer Documentation: Documentation providing customer information and initial hourly rate agreements.
3. Internal Documentation: Company related documentation, particularly repeatable processes conducted to solve a specific support event.
4. Procedure: details about a process or procedure that needs to be done repeatedly at different times.
5. Support Event: An amount of work done to resolve a problem, typically assigned by a ticket.
6. System: *White Wave Business Suite*
7. Ticket: A problem statement typically submitted by a user that authorizes a work event.

**2.2 User Characteristics**

The system will be used by technical support agents, Larry Dieterich, and potential interns of *White Wave Digital*. Through their field, the end users should have experience with technology and able to use the system with relative ease.

**2.3 Product Perspective**



* The *White Wave Business Suite* will receive inputs from the technicians and managers. These individuals will be able to access the system and provide data for support events, assets, financial information, and customer information. This information will be provided by the system to the database which will allow referral to the information within. The system will also provide relevant information to internal interns and customers.

**2.4 System Functional Requirements**

FR 1 Ticketing System which manages support events.

FR 2 Database for Technician, Client, and Asset information

FR 3 Generate and Send Invoices

FR 4 Track Payments and Purchases.

FR 5 Track Deliveries, Both Received and Made.

FR 6 Generate Internal Documentation

FR 7 Generate Customer Documentation

**2.5 General Constraints and Assumptions**

The system should be accessible from the field. The system will also be server based in order to provide a centralized location for technicians to access the database. The database will hold sensitive information that will be secured. Finally, the system will need to be cross platform and compatible with apple operating systems. The regular maintenance of data, including financial, asset, and ticket information, will be assumed to be maintained by the company and therefore not an automated feature within the system. Internet availability is also assumed for access between the system and the database.

**2.6 User View of Product Use**

The main page of the product will be a rectangular user interface. near the top of the interface, there will be several tabs, from which the user can access different parts of the database, such as financial information. On the left side of the window, there will be a window that holds all open tickets, that themselves can be opened to get more information about each ticket, as well as edit information or close the ticket. Inside the left window, will be a button that allows the creation of new tickets by filling in appropriate fields. A good description of it would be resembling a web browser with a special window on the left to hold tickets.

**3. Specific Requirements**

**3.1 Interface Requirements**

**3.1.1 User Interface**

The system will be based on a visual gui. Users will navigate menus and select options based on the tasks they wish the system to perform. Both the mobile and desktop versions of the application will be compatible with touch commands. Users will enter data into fields that accept input from a real or virtual keyboard.

**3.1.2 Hardware Interface**

A workstation connected to the internet with a working mouse, trackpad, or touch screen interface is necessary for the desktop version of the application. A smart phone with internet access is required for the mobile version.

**3.1.3 Software Interface**

Java-capable web browser with access to the internet, the Java Runtime Environment (JRE) from Oracle, or a smart phone compatible with ios or android.

**3.1.4 Communication Interfaces**

Internet access is required.

**3.2 Detailed Description of Functional Requirements**

**3.2.1 Template for describing functional requirements**

FR 1 Ticketing System which manages support events:

* Purpose: The ticketing system is a component of the system that will manage all support events. This will include the creation of the ticket, tracking of support events within the ticket, actions conducted in response to the ticket, time management tracking, closing of the ticket, editing within tickets, and technician information including their personal time allocation and actions taken.
* Inputs: Inputs into the ticketing system will comprise of two potential categories: technician information and support event information. Within technician information there will be inputs for their name [STRING] and ID information [STRING] as well as potential manual entry for personal time allocation [DOUBLE]. These inputs will be manually inputted by the user via text entry or through drop-down box selection. The support event information will encompasses several components. This category will have inputs for the original ticketing information and description [STRING]. It will also contain each individual support event’s comments which may contain a title [STRING], description [STRING], actions taken [STRING], and technician comments [STRING]. The system will also automatically input the time tracking information for the individual support events for the ticket.
* Processing: Upon processing, the system will display the available open tickets for the technicians to undertake. Inputs of technician information will be validated from an existing list of confirmed technicians. Upon confirmation the system will then proceed to track the time the technician spends upon the support events as well as any inputted descriptions or comments, saving them within the associated ticket information. When the ticket is closed, a finalization process will begin where the system totals the time spent upon the ticket for financial calculations as well as any final data saving associated with the ticket.
* Outputs: The ticketing system output will save all relevant information pertaining to the ticket in its individual object placed within a finalized ticket arraylist. At this finalization stage, the system will check if any required fields are missing or erroneous, such as technician name, support event names, or invalid timestamps.

FR 2 Database for Technician, Client, and Asset information

* Purpose: The database will hold all relevant information for technicians within the company, clients, and internal and external asset information. This database will be referenced throughout the system’s components allowing for both data validation and ease of access when information is to be retrieved.
* Input: The database will consist of employee information (ID, name) stored as strings, client information (name, company, location, phone number, location security comments) as strings, and asset information (name, model number, serial number, location, date of purchase, warranty information, associated tickets) stored as strings. This information will be inputted by the user manually.
* Processing: The system will create or edit existing items depending on the user selection. Data entry will be validated to verify that they are relevant to the corresponding input such as, phone numbers consisting of either 7 or 10 numbers.
* Output: The system will display database information within the GUI. New entries or edited entries will be saved to their respective arrays for ease of access in the future.

FR 3 Generate and Send Invoices

* Purpose: In order to show clients how much is owed for services rendered, part of this program will create invoices, and then store them for review by at least a standard user.
* Inputs: Invoices will need the amount owed, probably stored in either a Double or a Float, due to the ability to hold decimals. The location and date of work done will also be needed, both stored in Strings. The name of the company/client will be needed as well, stored as a String.
* Processing: After the file is created, A user must look it over to confirm everything is correct. If so, it will be sent.

Output: A file in PDF format will be created.

FR 4 Track Payments and Purchases:

* Purpose: Part of the program should be able to keep track of payments made to buy required products for the business (such as replacement computers, or a new carpet).
* Inputs: The needed inputs would be the product bought, the price of the product, the date of the purchase, the company/manufacturer of the product, and probably the purpose of the purchase. Most of these can probably be held in strings, although the amount paid can probably be held in a Float.
* Processing: The system will store these in a values in the database in such a way that they can be accessed by anyone able to use the database.
* Output: This system will be able to display the information to those who have access to the system.

FR 5 Track Deliveries, Both Received and Made:

* Purpose: Part of the program should be able to keep track of all packages sent and received by the company. This may be linked to the “Track Payments and Purchases” program.
* Inputs: For received deliveries, there would need to be the date received, probably in a string. There would also need to be name of the product received and quantity. For sending deliveries, there would need to be the company/client the product was sent to, stored in a string. Also, there would need to be the address for either the client/company. The date sent would be stored in a string, and the name of the product/whatever was sent would be stored in a string as well.
* Processing: The system will store these in a values in the database in such a way that they can be accessed by anyone able to use the database.
* Output: This system will be able to display the information to those who have access to the system.

FR 6 Generate Internal Documentation

* Purpose: The program will be able to generate internal information about the company’s current situation. This will give a general overview about important information like which employees are assigned to which work orders and more general financial information.
* Inputs: This function will pull data from the database. Required information will be employ names and IDs, the current number of open tickets, the status of work orders and asset data.
* Processing: The data will be formated to be useful to whoever requested the information. The total number of employees, assets, active work orders, and open tickets will have to be calculated at the time the request for an internal report is made.
* Output: The request will be filled when a document is generated listing useful information about the status of internal operations in a PDF style format.

FR 7 Generate Customer Documentation

* Purpose: This function is able to generate a document containing information about a specific customer, including invoices and information about the specific tickets and support events associated with a customer and their company.
* Inputs: Required data is taken from the database and includes customer name and the current support events associated with that customer. Dates about the start times and end times of support events and tickets and the hourly rate agreed upon are also necessary.
* Processing: The data is processed so it is more helpful in the description of all information about the client. Total numbers of support events and the number of hours spent on those events is calculated. The hourly rate is used along with the previously calculated information is used to generate an accurate invoice for the specific customer.
* Output: The request is completed when all available information about the client is generated and presented in the relevant document in a PDF style format.

**3.2.2 Data Dictionary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Name** | **Type** | **Where Used** | **Comments** |
| 1 | Technician name | varchar | Employee Table |  |
| 2 | Is Admin | bool | Employee Table |  |
| 3 | Technician ID | unsigned int | Employee Table | Auto Increment |
| 4 | Client Name | varchar | Client Table |  |
| 5 | Client Phone Number | varchar | Client Table |  |
| 6 | Company Name | varchar | Client Table |  |
| 7 | Company Comments | varchar | Client Table | Any comments concerning access to their facility (i.e. security codes) |
| 8 | Company ID | unsigned int | Client Table | Don’t Auto Increment. Is assosiated with a client and not on its own |
| 9 | Client ID | unsigned int | Client Table | Auto Increment |
| 10 | Hourly Rate | float | Client Table |  |
| 11 | Ticket Name | varchar | Ticket Table |  |
| 12 | Ticket Time Start | timestamp | Ticket Table | Can be applied to both the total ticket time and individual support event times |
| 13 | Ticket Time End | timestamp | Ticket Table | Can be applied to both the total ticket time and individual support even times. |
| 14 | Ticket Total Time | unsigned int | Ticket Table | Will contain the final calculation for time spent on a specific ticket |
| 15 | Ticket ID | unsigned int | Ticket Table | Auto Increment |
| 16 | Support Event Name | varchar | Support Event Table | Hourly fee for support as dictated by customer agreemen |
| 17 | Support Event Comments | varchar | Support Event Table | Comments or procedures used in the support event |
| 18 | Support Event ID | unsigned int | Support Event Table | Auto Increment |
| 19 | Asset Name | varchar | Asset Table |  |
| 20 | Model Number | varchar | Asset Table | Asset model number |
| 21 | Serial Number | varchar | Asset Table | Asset serial number |
| 22 | Asset DoP | date | Asset Table | Date of Purchase |
| 23 | Warranty Information | varchar | Asset Table | Warranty expiration date |
| 24 | Asset Location | varchar | Asset Table | Customer location or internal location |
| 25 | Asset Price | float | Asset Table | Initial purchase price of asset |
| 26 | Asset cumulative price | float | Asset Table | Total price of asset (including all repairs/service) |
| 27 | Asset ID | unsigned int | Asset Table | Auto Increment |

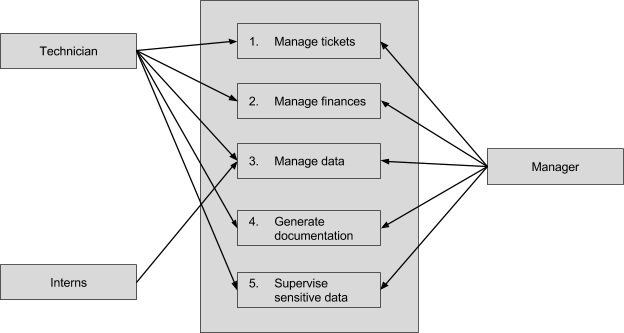
**3.3 Non-Functional requirements**

Basic security is to be provided, especially in protecting the stored passwords within the database. The interface must be user-friendly, allowing efficient entry and retrieval of information and data. System must be reliable and able to maintain the database of information and ticket information without unintentional loss of data. Also, the system must be accurate in it’s calculations of financial statements.

**4. Object Oriented Analysis (OOA) -UML**

**4.1. Object Oriented Analysis**

**4.1.1** **Use Case Diagram.**



**4.1.2** **Use Case Descriptions**

Use Case Name: Manage Tickets

Use Case Number: UC1

Authors: Austin Kerbow

Actors: Technician, Manager

Overview: This use case involves the creation and editing of tickets

References: FR1

Related Use Cases: Manage Data

Typical Flow:

Precondition: Assume no one is logged into the current running client program

1. User enters login credentials
2. Verify user credentials
   1. If the user enters invalid credentials **return to step 1**
3. User selects the option to create a new ticket
4. The user enters all the relevant data for the ticket including the name of the ticket, and a comment about its purpose
   1. If not all entered data is valid or a required field is left empty **return to step 4**
5. The ticket is created and all relevant data is stored in the database, the time the ticket was created is recorded

Post-condition: If data is stored correctly the user is returned to the main menu

Alternative Flow:

Precondition: Assume no one is logged into the current running client program

1. User enters login credentials
2. Verify user credentials
   1. If the user enters invalid credentials **return to step 1**
3. The user selects the option to edit an existing ticket
4. The user searches for the name of a ticket and the search results are listed
5. The user selects a ticket from the list
6. The user is shown the existing fields for the selected ticket
   1. The user can edit any of the fields
   2. The user can mark the ticket as completed
      1. The ending time for the ticket is recorded and the total time the ticket was open is calculated
   3. If not all data was valid, or a required field was left empty **return to step 6**
7. The edited ticket information is stored in the database

Post-condition: If data is stored correctly the user is returned to the main menu

Use Case Name: Manage Finances

Use case Number: UC2

Authors: Scott Kennedy

Actors: Technician, Manager

Overview: This use case outlines operations like viewing and editing financial information

References: FR3, FR4

Typical Flow:

Precondition: User is not logged in and using the system.

1. User enters login credentials.

1. Verify user credentials.

2.1. If the user enters invalid credentials **return to step 1.**

1. User selects “Finances” tab.
2. User selects option to create an invoice.
3. User enters all required and relevant information for the creation of the invoice.

5.1. If information is invalid, or required fields are left blank, **return to step 5.**

1. An invoice is created in pdf format.

6.1. A timer is set to remind creation of another invoice in a month.

6.2. A Manager should review the invoice before sending. If it is incorrect, **return to step 5.**

1. The invoice is stored in the database for later reference.

Postcondition: if data is stored correctly, User is returned to Main tab.

Alternate Flow:

Precondition: User is not logged in and using the system.

1. User enters login credentials.

1. Verify user credentials.

2.1. If the user enters invalid credentials **return to step 1.**

1. User selects “Finances” tab.
2. User selects option to Manage Deliveries
3. User enters all required and relevant information for the Delivery Log.

5.1. If information is invalid, or required fields are left blank, **return to step 5.**

1. The Log is updated and saved in the database.

Postcondition: if data is stored correctly, User is returned to Main tab.

Use Case Name: Manage Data

Use Case Number: UC 3

Authors: Austin Kerbow

Actors: Technician, Manager, Intern

Overview: View non-sensitive data currently stored in the database

References: FR 2

Related Use Cases: UC1, UC2

Typical Flow:

Precondition: User is actively using the system

1. The user selects what data they wish to view
   1. User selects the option to view tickets
      1. The user will enter the name of the ticket they want to search for
      2. The user selects the relevant ticket from a list and views the data
   2. User selects the option to view non-sensitive data
      1. The user will enter a search for the data they want to view
      2. The user selects the relevant data from the search
2. The active user indicates they are finished viewing the data

Post-Condition: User is returned to the main menu

Alternate Flow:

Precondition: User is actively using the system

1. The user selects what data they wish to view
2. The user tries to view data they are not authorized to view
3. They are prompted to either login as a user with sufficient privileges or view non-sensitive data

Post-Condition: User is returned to the main menu

Use Case Name: Generate Documentation

Use Case Number: UC 4

Authors: Erik Hsu

Actors: Technician, Manager

Overview: This use case outlines the generation of financial and customer documentation.

References: FR 3, FR6, FR7

Related Use Cases: UC1, UC3

Typical Flow:

Precondition: Necessary data is available within the database and the user is actively using the system

1. User selects the financial or customer documentation option.
   1. Financial Option
      1. User will select a customer with closed tickets.
      2. System will pull relevant timestamp and hourly rate information from the ticket and initial customer contract located within the database.
      3. The system will calculate the invoice for the tickets completed for the pay period.
      4. The system displays the completed document for the user.
      5. The document is saved.
      6. The user is displayed an option to print the invoice.
      7. User exits the financial section.
   2. Customer documentation option.
      1. User selects “new customer”.
      2. User will be provided with necessary fields to be populated for all client information.
      3. Based upon the inputted data, the system will save and provide a document listing the relevant information for client reference in the future.
      4. The user is displayed an option to print the invoice.
      5. User exits the customer documentation section.
2. User exits the documentation section.

Postcondition: The system returns the interface to the main menu.

Alternate Flow:

Precondition: User is actively using the system.

1. User selects the financial or customer documentation option.
   1. Financial Option
      1. User will select a customer with closed tickets.
      2. System will attempt to pull relevant timestamp and hourly rate information from the ticket and initial customer contract located within the database.
      3. System is missing information to proceed.
      4. User is notified what information is missing.
      5. User corrects the missing information through UC1 and UC3
      6. User re-enters the financial option and begins from step 1.1.1.
      7. The system is able to successfully generate the financial document and proceeds to follow typical flow.
   2. Customer documentation option
      1. User selects “new customer”.
      2. User will be provided with necessary fields to be populated for all client information.
      3. User attempts to save and finalize.
      4. System finds invalid or missing data and an error prompts the user.
      5. User is returned to step 1.2.2.
2. User exits the documentation section.

Postcondition: The system returns the interface to the main menu.

Use Case Name: Supervise Sensitive Data

Use Case Number: UC5

Authors: Erik Hsu

Actors: Technician, Manager

Overview: This use case outlines the access and editing of sensitive data.

References: FR2

Related Use Cases: N/A

Typical Flow:

Precondition: User is actively using the system.

1. User presses the button to enter the sensitive data section.
2. Pop-up notification appears for login and password information.
3. User enters login credentials.
4. Login information is verified and access is granted.
5. User selects customer or company where data is to be retrieved.
6. Interface displays sensitive data
7. User exits the sensitive data section.

Postcondition: System relocks the sensitive data, process needs to be repeated to access again.

Alternate Flow:

Precondition: User is actively using the system.

1. User presses the button to enter the sensitive data section.
2. Pop-up notification appears for login and password information.
3. User enters login credentials.
4. Login information is deemed invalid.
5. System closes the sensitive data section.

Postcondition: System relocks the sensitive data, process needs to be repeated with valid login information to proceed.

**4.1.3** **List potential / analysis classes based on the problem statement and use case diagram.**

* N/A

**5. Special Remarks or Comments**

Specifications are adaptable to change based upon client necessities and requests.

**6. References or Resources Used**

**7. Team member’s Roles and Approval**

**7.1. Team member’s roles in completing the SRS**

Erik Hsu

* Section 1
* Section 2 (2.1, 2.2, 2.3, 2.4)
* Section 3 (3.1, 3.2.1, 3.2.2, 3.3)
* Section 4 (4.1.1, 4.1.2)

Scott Kennedy

* Section 2 (2.4, 2.5, 2.6)
* Section 3 (3.1, 3.2.1)
* Section 4 (4.1.1, 4.1.2)

Austin Kerbow

* Section 2 (2.4)
* Section 3 (3.1, 3.2.1, 3.2.2)
* Section 4 (4.1.1, 4.1.2)

**7.2. Team member signatures and date signed**

Erik Hsu Scott Kennedy Austin Kerbow