**91.411.201: Software Engineering I**

**Software Requirements Specification**

**91.411.201: Software Engineering I**

**(Team Number)**

Muhammed Khalid (@mkhalid578)

Vibhuti Patel (@vibhutipatel18)

**(Order Processing System)**

**Software Requirements Specification**

**Document**

**Version: (01.00)** **Date: (04/03/2017)**

**Table of Contents**

1. Introduction 51.1 Purpose 5

1.2 Scope 5

1.3 Definitions, Acronyms, and Abbreviations 5

1.4 References 5

1.5 Overview

5

2. The Overall Description 6

2.1 Product Perspective 6

2.1.1 System Interfaces 6

2.1.2 Interfaces 6

2.1.3 Hardware Interfaces 6

2.1.4 Software Interfaces 7

2.1.5 Communications Interfaces 7

2.1.6 Memory Constraints 7

2.1.7 Operations 7

2.1.8 Site Adaptation Requirements 7

2.2 Product Functions 8

2.3 User Characteristics 8

2.4 Constraints 8

2.5 Assumptions and Dependencies 9

2.6 Apportioning of Requirements

9

3. Specific Requirements 9

3.1 External interfaces 10

3.2 Functions 10

3.3 Performance Requirements 11

3.4 Logical Database Requirements 11

3.5 Design Constraints 12

3.5.1 Standards Compliance 12

3.6 Software System Attributes 12

3.6.1 Reliability 12

3.6.2 Availability 12

3.6.3 Security 12

3.6.4 Maintainability 13

3.6.5 Portability 13

3.7 Organizing the Specific Requirements 14

3.7.1 System Mode 14

3.7.2 User Class 14

3.7.3 Objects 14

3.7.4 Feature 14

3.7.5 Stimulus 15

3.7.6 Response 15

3.7.7 Functional Hierarchy 15

3.8 Additional Comments

15

4. Change Management Process

5. Document Approvals

6. Supporting Information 15

# 1. Introduction

The product will behave a system to help users in a company to place orders quick and efficiently. Further help users by allow order approvals, and avoid redundant orders to maximize companies budget and to avoid excessive spending.

## 1.1 Purpose

This product is prefaced to be used by any company in which one needs to order materials from a third party retailer. Ideally places such as schools, startups, corporations, and medical devices and healthcare will be our ideal clients.

## 1.2 Scope

*In this subsection:*

1. A company login which will be produced for each client
2. Followed by a user login
3. Anyone within a company can place orders
4. Orders go through an approval stage and are approved by each manager of a department.
5. Orders that are approved, will be marked approved and available for all users to see
6. Managers can view all pending orders
7. Users can view all approved orders

## 1.3 Definitions, Acronyms, and Abbreviations.

*Provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS.*

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Python | A high-level objected oriented programming and scripting language |
| Flask | A python web framework used for rapid back-end web development |
| JavaScript | A client side web programming language. |
| HTML | Hypertext Markup Language used for Front Side web development and GUI design |
| CSS | Cascading Style Sheets. These are used for adding “prettiness” to HTML |
| MySQL | An open source and free SQL-based database created by Oracle |
| Google Cloud | A paid hosting platform provided by Google. |
| SQL | Structured Query Language. A language used for reference and inserting data into a SQL-based database. |
| GUI | Graphical User Interface |

## 1.4 References (Omitting)

*We have not go far enough in our project to refer to other documents. This our first Software Requirement Specification.*

## 1.5 Overview

*In this subsection:*

1. The Overall Description
2. Product Perspective
3. Interfaces
4. Constraints
5. Operations
6. Product Functions
7. Product Dependencies

# 2. The Overall Description

This product is a web application used for placing orders within a particular company. It will give users within any department to place orders and also see orders the department has already placed, which will help in avoid unnecessary spending. Users can vary from normal users such as regular employees and users who are manager and directors. These users will have an additional feature which will allow them to approve and deny orders within their department. They can also see but not modify orders placed in other departments.

## 2.1 Product Perspective

There is a product that exists which shares similarities with our product. Moreover, we used to this as a reference to this application. The name of the application is call Candex (<https://www.candex.com/>). The product is independent and self-contained. In other words, this product does not require any additional dependencies other than the dependencies mentioned in this document. We are not using any third party retailers and the technologies that we are implementing in this product are open source.

Candex is our fictional competitor. They are a real company and they also so have the same design as our company. Candex also offers invoice notification between buyers and sellers. Candex offers a stand alone application which always behaves as a Software as a Service (SaaS). Our application is a web application and does not share the same complexity as Candex. Candex is also intended more for corporate companies to handle invoices with outside sellers. Our is product is more limited to the company itself. We are not providing a relation between a company and their respective sellers, we are providing the company a more organized and efficient approach to handling their invoices. We want individuals in a company to avoid excessive spending by referring to our software. We do not care who they buy from.

Basic Architecture/ Layout:

Order Window

Comp Login

User Login

*The following subsections describe how the software operates inside various constraints*.

### 2.1.2 Interfaces

The user interface has the following constraints:

* Company Login – Users will login to the application with set of company confidential credentials
* User Login – Users will then be prompted with a user login page using credentials provided by their company for their user profile
* Main Order Window – This is the page that will presented the user upon a successful login. They will be able to see pending orders within their department, and they will be able to place their own orders which will require approval from their respective department managers and/or directors.

### 2.1.3 Hardware Interfaces

The hardware systems have the following interface constraints:

* The MySQL database only has 10GB of storage. Storage cannot beyond the space reserved
* No QPS (Queries Per Second) limit. However, the size and connection limits are present. The server requires a network IP to access the MySQL database. Companies will need to provide their company’s public IP to gain access to the database.
* Disk speeds provided by Google Cloud. The speed at which each request from the server can only be handled by the speeds provided by our host.

### 2.1.4 Software Interfaces

The software interface has the following constraints:

* Python 2.7 is the version of python that we are using. Any newer version of python (3.xx or above) may provide issues with compilation and interpretation. We urge users and developers to use Python 2.7 to avoid any issues
* MySQL 5.7 is the version of MySQL being used in this application. Developers must stick to this version of MySQL otherwise issues could arise with queries and insertion statements.
* HTML5 – We are using the latest version of HTML and CSS technologies were urge developers to stay up to date with HTML (i.e HTML5 or higher).
* Google Chrome – Please use Google Chrome to run application as it is the most versatile and stable when running our web applications. Developers have faced issues with Mozilla Firefox and other browsers. Any version of Google Chrome works fine.

### 2.1.5 Communications Interfaces

*Using Google Cloud platform to handle web services*

### 2.1.6 Memory Constraints

*No Memory Constraints*

### 2.1.7 Operations

Daily backups occur between 10:00 and 2:00 pm. Users are welcome to continue using the applications as this will not interfere with their work.

### 2.1.8 Site Adaptation Requirements (omitted section)

*In this section:*

1. *Define the requirements for any data or initialization sequences that are specific to a given site, mission, or operational mode*
2. *Specify the site or mission-related features that should be modified to adapt the software to a particular installation*

*If any modifications to the customer’s work area would be required by your system, then document that here. For instance, “A 100Kw backup generator and 10000 BTU air conditioning system must be installed at the user site prior to software installation”.*

*This could also be software-specific like, “New data tables created for this system must be installed on the company’s existing DB server and populated prior to system activation.” Any equipment the customer would need to buy or any software setup that needs to be done so that your system will install and operate correctly should be documented here.*

## 2.2 Product Functions

**Invalid Password** – The user has entered a correct username, however the password associated with that user is incorrect

**Invalid Username** – The user has entered an incorrect username or the username does not exist

**User already exists** – The user has attempted to register a user that already exists

**Company already exists** – The user has attempted to register a company that already exists

**Register New User** – A button that will prompt a new window to allow a new user to register within the application.

**Register New Company** – Allow a user to register a new company with the order processing system.

**Form** – An input form which will allow a user or manager to place an order

**Order Data** – All orders that have been placed and are available for users to see. Users are limited to see orders in their department and orders they have already placed. Managers are able to see orders placed in their department, orders they have placed, and orders placed by other users in separate departments.

**Profile** – Each user will have a custom profile which he or she may modify as he or she pleases.

**Rejected** – An order that has been refused by a manager and/ or director.

**Approved** – An order that has been approved by a manager and/or director.

**Logged In** – Let the user know they have successfully logged in.

**Logged Out** – Let the user know they have successfully logged in

## 2.3 User Characteristics

The user is expected to be a valid employee of a company and have a small technical background in data entry. They should be familiar with browsing the web and basic computer use. The user should also be capable of remember his or her password to login.

## 2.4 Constraints

The application is currently hosted locally on a computer. Users will not be able to access a product version of this web application just yet.

Database size is limited. Currently the database is only large enough to support up to 10GB of raw data. Currently, 1.8 GB are being used to run the database with the schemas that we have provided for the database layout.

The application will also constrained by the user’s computer. We recommend at least 4GB of storage and 128GB of hard drive space for optimal performance

## 2.5 Assumptions and Dependencies

A valid internet connection because we are using a database that is in the cloud and not locally hosted. Without a valid internet connection, the application will not work at all.

The user is familiar with basic computer use and data entry. They should feel comfortable logging in and out, knowing how to type with a keyboard and click with a mouse.

## 2.6 Apportioning of Requirements.

A member of the company has left and that may hinder our future plans. Currently, everything is going to how we would like it to go.

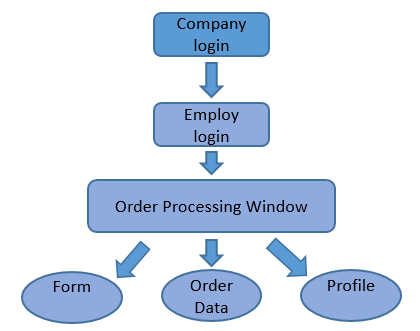
We are not sure if we are able to complete a full login system which will maintain user sessions and allow users to modify their own profiles. This is a quite a jump but we are optimistic and hope to complete by the projects official deadline.

# 3. Specific Requirements

This section contains all the software requirements at a level of detail sufficient to enable designers to approach the design of a system to satisfy those requirements and test engineer to test that the system satisfies those requirements. This order processing application shall run in a web browser a consist of a graphical user interface and a database. Also, this application system shall be able to allow multiple user to log in at the same time.

## 3.1 External Interfaces

The user friendly graphical interface should be created using HTML, JavaScript, CSS or Python languages. The user shall be able to log in using company account and then employ account information. Once user will be logged in, then he or she shall able to see all the order details which is placed by company and also be able to fill the digital form for requesting new order. Also there are three kinds of user who has different abilities to use this application in meaning full way. The basic architecture design of interfaces is below. And a detailed description of all inputs into and outputs from the application is following after diagram:



**3.1.1 Company Log in Page**

The main purpose of the company login account is that company’s employees can only have access to their company information. So to log in with company account information, a system shall have company login page and create new login company account page. This login page shall take two text inputs, first input shall be company user id and second company account password. User needs to enter correct data to login. Characters shall be masked in password input box and password shall be case sensitive. If user will not have company account, then a system shall provide option to create new one. For creating new account, system shall take company user id and password as inputs and store it in database. Once new company account will be created then system shall allow use to login with new account. If the company account exists in database, then system shall not create new account.

**3.1.2 Employ Log in Page**

The main purpose of the employ login account is that company’s different positions employees can have different powers to edit order details. So to log in with an employ account information, a system shall have employ login page and create new employ login account page. This login page shall take two text inputs, first input shall be employ user id and second employ account password. Characters shall be masked in password input box and password shall be case sensitive. User needs to enter correct employ id and password as input to login. If user will not have an employ account, then a system shall provide option to create new one. For creating new account, system shall take first name, last name, employ email ID, Employ position in company, department, employ user id and password as inputs and store it in database. During new account for employ, system shall ask whether employ is order handler, manager or normal employ in the company. This question shall be asked because employees with different position shall have different authorities for order processing work. Once new employ account will be created then system shall allow use to login with new employ account. If the employ account already exists in database, then system shall not create new account.

**3.1.3 Main Processing Window**

Once, user will log into employ account then he or she shall see the main processing window of a system. There shall be three navigation tabs on the window with in order like: Form, Order Data and Profile. Also there shall be “logout” button to log out from account. When user will log out, then Use shall be able to login again by entering company login information. The form tab shall be active then user log in.

**3.1.4 Order Form tab**

The purpose of the Order form is to send request for new order including item details. Basically, this page shall be containing order form with some input text boxes. The order form shall take requesting person’s first name, last name, email id, position, department as inputs. Also the form shall take data of item-name, item-detail, item-quantities, place (from where) information, time-period and why it is wanted as inputs. All inputs shall be required to fill. Then there shall be submit button at the end to send a request of new order with all the information.

**3.1.5 Order Data tab**

This tab shall be listing of all the orders that exist for the company in database. There shall be nineteen columns for different field in the table like (in order): item-number, first-name, last-name, email-id, position, department, item-name, item-detail, item-quantities, place (from where) information, time-period, use/reason, placed order data comment, order-status, shipment-company name, tracking number of shipment, tracking website name, expected arriving date, and arrived date. But item-number, email-id, position, item-detail, place (from where) information, time-period, placed order data comment, shipment-company name, tracking number of shipment, tracking website name, expected arriving date, and arrived date columns shall be invisible in table. All of these inputs in table shall have text data type. Also when user select any row with containing order data in table, that time dialog box shall be popup. This dialog box shall show all the information about that item order including invisible columns data. This dialog box shall have two button in bottom for editing order information and storing the updated order data. User shall be able to edit the information of order status, shipment company, tracking number, tracking website name, expected arriving date and arrived date by clicking the edit button on dialog box. To store the update data in database, user has to select save button in bottom of pop up dialog box.

**3.1.6 Profile tab**

This tab shall be containing the user information who has logged in. Also user shall be able to edit the information if he or she will want. There shall be edit button to allow user to edit the profile information. There shall be save button to allow user to save new edited data in database for future use. The profile tab shall have information about user’s first name, last name, position, department, user id, password, email id, and information on whether user is order handler, manager or employ. Characters of password shall be masked.

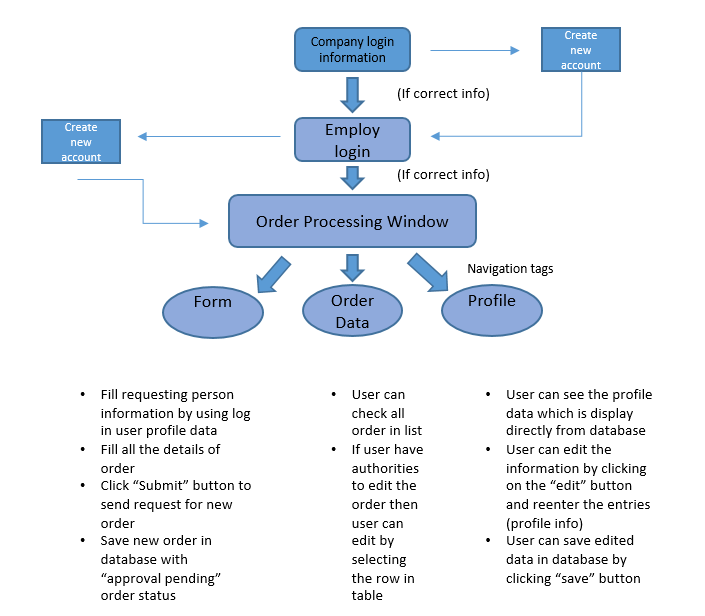
## 3.2 Functions

The database of company accounts, employ accounts and order information shall be used for back end work. A system shall allow char data type data to store in database. All data in database shall be in char data type. The system shall start with company login page. User has to enter the correct company user id and password to log into company account. When user shall click on “log in” button, then system shall compare the given company user id and password with the company account data in database. If the inputs will not match, then system shall not allow user to log in. If user will need new company account, then system shall allow user to click on “register new company account” button and allow to create new one. When user will select the “create account” after entering information about new account of company, that time system shall check into database if account would have been existing. If company account name will exist, then system shall not allow to create new one. And if company account name will not exist, then given new company user id and password shall be stored in database and user shall be allowed to log in. One user will log in to company account then system shall ask information for employ log in.

For logging into employ account, user has to enter the correct employ user id and password. A function shall check the given inputs with existing data in database for log in process. If user will need new employ account, then system shall allow user to click on “register new employ account” button and allow to create new one. When user will select the “create account” after entering information about new employ of company, that time system shall check into database if account would have been existing. If employ account for that company user id will exist, then system shall not allow to create new one. And if employ account user id will not exist, then given new employ information shall be stored in database and user shall be allowed to log in. Once user will log in to employ account then system shall show processing window with three navigation tabs like: form, order data and profile.

Once user will log in with employ account information, system shall display employ (logged in) information in profile tab on procession window. User shall be able to edit that information by clicking on edit button and change the account information. But to change the information in database for future use, User has to click on save button. After click on save button, system shall save the updated information in database.

User shall be able to fill the order form to request new order. The requesting person information shall be filled by using employ (logged in) profile information so user will not need to enter that information. A system shall take all the information about order otherwise system shall not allow user to submit the order requesting form. When user will click on submit button in the bottom of the form, the function in back end shall take all the inputs related to order and store in database with order status “approval pending”. At the same time, system shall be notifying manager of requesting person’s department to approve a new order. Now, new order shall be available in database so user shall be able to see the requested order in order data table in order data tab of the window. The basic functionality of the system is shown below:

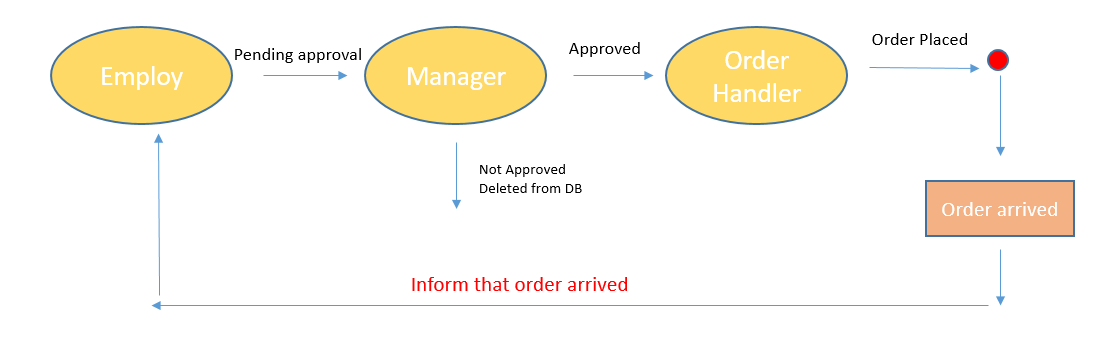


Manager shall be able to approve the order by editing the order status of that requested new order. To make changes for order, manager shall log in with his or her employ id and check the order in order list in the order data tab in processing window. Manager shall select the row which shall containing detailed data of order. When manager will select the row, the dialog box shall be pop up with all the information in content window of box. Manager shall be able to edit the order status of order on dialog box window by click on edit button on dialog box. When manager will hit the submit button on dialog box, order data shall be updated in database with given new inputs.

When manager will approve the order, the order handler person shall be notifying to place new order. If manager will not approve the order, then system shall be notifying requesting person and system shall delete order from database. For placing the order, order handler shall be able to check the order in order data table and select the order to get all the information related to that order. After placing the order, order handle shall be able to change the order status and able to enter the shipping information by clicking edit button on pop up dialog box. When order will be placed, the person who requested that item shall be notify. A system shall use database to get email id for notifying employ, manager and order handler.

When order will have arrived in shipment, then order handler shall be able to edit the arriving date and status of the order. A system shall be notifying the requesting person that order arrived.

Thus, only one user who is the order handle of the company shall have the ability to edit the order status and enter the shipping information of the order. The manager shall be able to edit the order status of order which have been requested from his or her department only. The manager from one department shall not be able to edit other’s department order status. And the employ of the company shall not able to edit the order at any time. So all the function shall take the login user information to give ability of changing data. The notifying functionality is shown below in diagram:

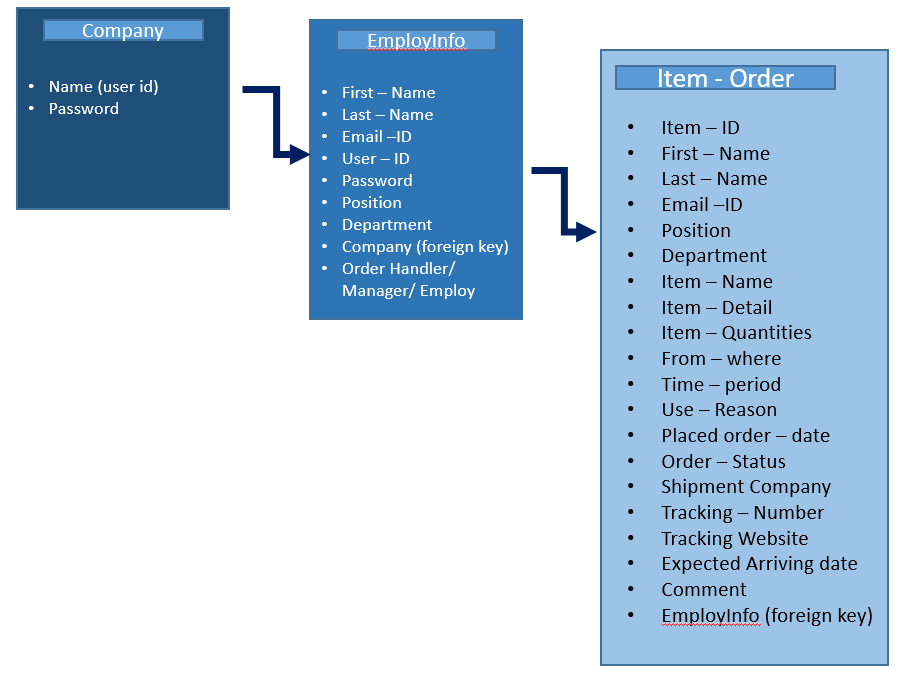


**3.3 Performance Requirements**

## This system shall not have complication functionality, so 97% of the transactions shall be processed in less than 1 second. A system shall need the internet connection so very slow internet speed will affect the performance of a system. But system shall be able to perform fast with standard internet speed. A system shall upload data on display using back end database work. But 98% of the transactions from database shall be processed in less than 1 second to upload data in a system.

## 3.4 Logical Database Requirements

The database structure diagram is shown below which is containing Data entities and their relationships:



A system shall have database with three tables and shall have foreign key connection as shown in the diagram. A system shall use the data from company database table for allowing user to log in or creating new company account. Same way a system shall use the data from employInfo database table for allowing user to log into employ account or creating new employ account under logged in company name. A system shall access information from Company and employinfo table only at log in process.

A system shall upload order list in the table from Item–Order database table. Also system shall update the information in respective data table using (company->name), (employ->user id) or (item-order-> item-number) information. A system shall not need to upload data from database very frequently. That’s why system shall be able to perform without any delay process.

## 3.5 Design Constraints

System shall give limitation for user to work. This may seem backward at first but, by understanding and implementing this constraint when designing and building system will help users navigate and use system with minimal errors, thus creating a more satisfactory user experience.

The Internet connection is also a constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function.

**3.6 Software System Attributes**

The three most important quality criteria for Web application success (and thus, the underlying software) as

1. reliability,

2. portability, and

3. security.

Additional important criteria include

4. availability, and

5. maintainability

### 3.6.1 Reliability

The reliability that the system allows the user to login with both company and employ account information. Also system allows user to fill the form, edit the profile information, and check the order data in table. User with authorities is allowed to edit the order information and status of the order by the system. System must work smoothly without any delay. System must only allow editing authority to manger or order handler user. Multiple user should be able to login at same time. Order data or account information should be updated on database with in 1 second. If user is not using the application system, the system must log out the user for security.

### 3.6.2 Availability

A system must log out user if any error accurses. The system should be connected to the internet in order for the application to communicate with the database. A system should be usable 24/7.

**3.6.3 Security**

A user should not be able to log-in for a certain time period after three times of failed log-in attempts of company account. A user should not be able to log-in for a certain time period after four times of failed log-in attempts of employ account. If user is not using the application system for a certain time period, then the system must log out the user for security.

**3.6.4 Maintainability**

The application should be easy to extend. The code should be written in a way that it favors implementation of new functions. Test environments should be built for the application to allow testing of the applications different functions in order to test the system.

### 3.6.5 Portability

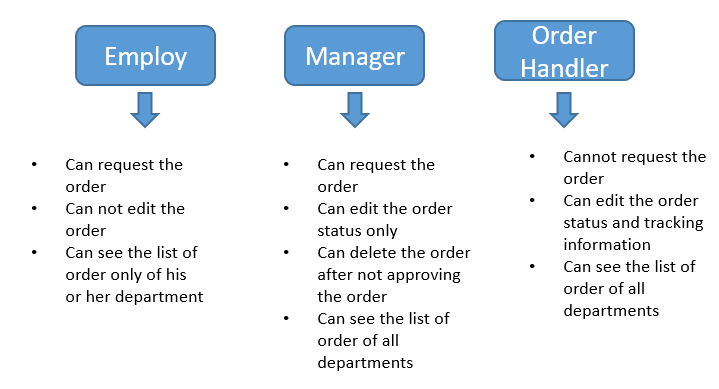
The application should be portable with any Browser and on any operating system. System will be tested on different browsers and on different operating systems.

## 3.7 Organizing the Specific Requirements

### 3.7.1 System Mode (not needed)

### 3.7.2 User Class

A system shall have three kinds of users and they will have different powers which are shown in diagram below:



### 3.7.3 Objects

### 3.7.4 Feature

A system shall have feature like notifying person by email id for giving order process information. A system shall be notifying manager when new order requested from his or her department. If the manager will approve the order, then system shall be notifying order handler and requesting person that order approved. When order handler will change the order status, that time system shall be notifying requesting person about order status information.

### 3.7.5 Stimulus

### 3.7.6 Response

3.7.7 Functional Hierarchy (Omitted)

## 3.8 Additional Comments

*Whenever a new SRS is contemplated, more than one of the organizational techniques given in 3.7 may be appropriate. In such cases, organize the specific requirements for multiple hierarchies tailored to the specific needs of the system under specification.*

*Three are many notations, methods, and automated support tools available to aid in the documentation of requirements. For the most part, their usefulness is a function of organization. For example, when organizing by mode, finite state machines or state charts may prove helpful; when organizing by object, object-oriented analysis may prove helpful; when organizing by feature, stimulus-response sequences may prove helpful; when organizing by functional hierarchy, data flow diagrams and data dictionaries may prove helpful.*

*In any of the outlines below, those sections called “Functional Requirement i” may be described in native language, in pseudocode, in a system definition language, or in four subsections titled: Introduction, Inputs, Processing, Outputs.*

# Change Management Process

*Identify the change management process to be used to identify, log, evaluate, and update the SRS to reflect changes in project scope and requirements. How are you going to control changes to the requirements. Can the customer just call up and ask for something new? Does your team have to reach consensus? How do changes to requirements get submitted to the team? Formally in writing, email or phone call?*

The process that will be using to account for any changes in the SRS will start with identification and then we will log it. After we have logged the issue or the change, we will evaluate within a sprint cycle and update to reflect changes in project scope and requirements.

The changes to the requirement will occur based of customer experience and deficiencies found by the developers. Ideally, we will tackle it with in a sprint meeting where we can have a Test Engineer provide a SRS validation. We will then conduct a separate meeting in which we evaluate the changes as a team and have a member make the changes and perform a dry run and then a formal validation before it is used for production.

Customer comments will be logged via phone calls and email. We will update the SRS based off what the user would like to see in the product. Developers are very biased, while customers are not.

# Document Approvals

Approved by: Dr. Yu Cao

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Supporting Information

* 1. Candex [www.candex.com](http://www.candex.com)
  2. GitHub Repository [www.github.com/mkhalid578/sw\_eng\_project](http://www.github.com/mkhalid578/sw_eng_project)
  3. Docs: [www.github.com/mkhalid578/sw\_eng\_project/tree/master/Documentation](http://www.github.com/mkhalid578/sw_eng_project/tree/master/Documentation)