CS532 Software Engineering Project Notebook

ICE TRACK System

Professor Leonard

February 18, 2016

Team ICE TRACK

Dustin Bonilla

Eric Ly

Ali Saadati

Mohammed Alshahrani

Anass Benothmane

Colan Kirk

**Team Plan**

1. We chose Agile as the life cycle model for our project. Our reasoning for using this model is that the system is not extremely critical, since any errors in the systems would not cause major harm to anybody; the complexity of the system is rather low, mainly consisting of database lookups and simple GUI controls. The development environment will use no COTS (Ruby on Rails, SQLite). Key aspects of this life cycle model include evolutionary development, welcoming and quickly responding to changes in requirements, close cooperation between client and development team, minimal bureaucracy, and most importantly frequent communication among team members. The advantage to using this life cycle model will be that there will be considerably less documentation and overall “fluff” while building the system. This model also allows the development team to keep the user in mind throughout the entire project because there will be ongoing communication between the team and the user.
2. **See attached Spreadsheet: Schedule v1.0**
3. So far, everything has gone according to plan and we are on schedule. We estimated 4 labor hours per module for requirements and 14 per module for design and we have totaled about 20-30 labor hours as of the end of the requirements phase, going into the design phase.
4. **See attached Spreadsheet: Schedule v1.0**

**Requirements**

1. Front End Requirements:
   1. Main/ Login screen interface
   2. Admin only screen to add, change and remove users
   3. Inventory management interface
   4. Shipment tracking interface
   5. Order entry screen interface
   6. Trouble ticket management screen interface
   7. M-V-C implementation to connect back end with front end

Back End Requirements:

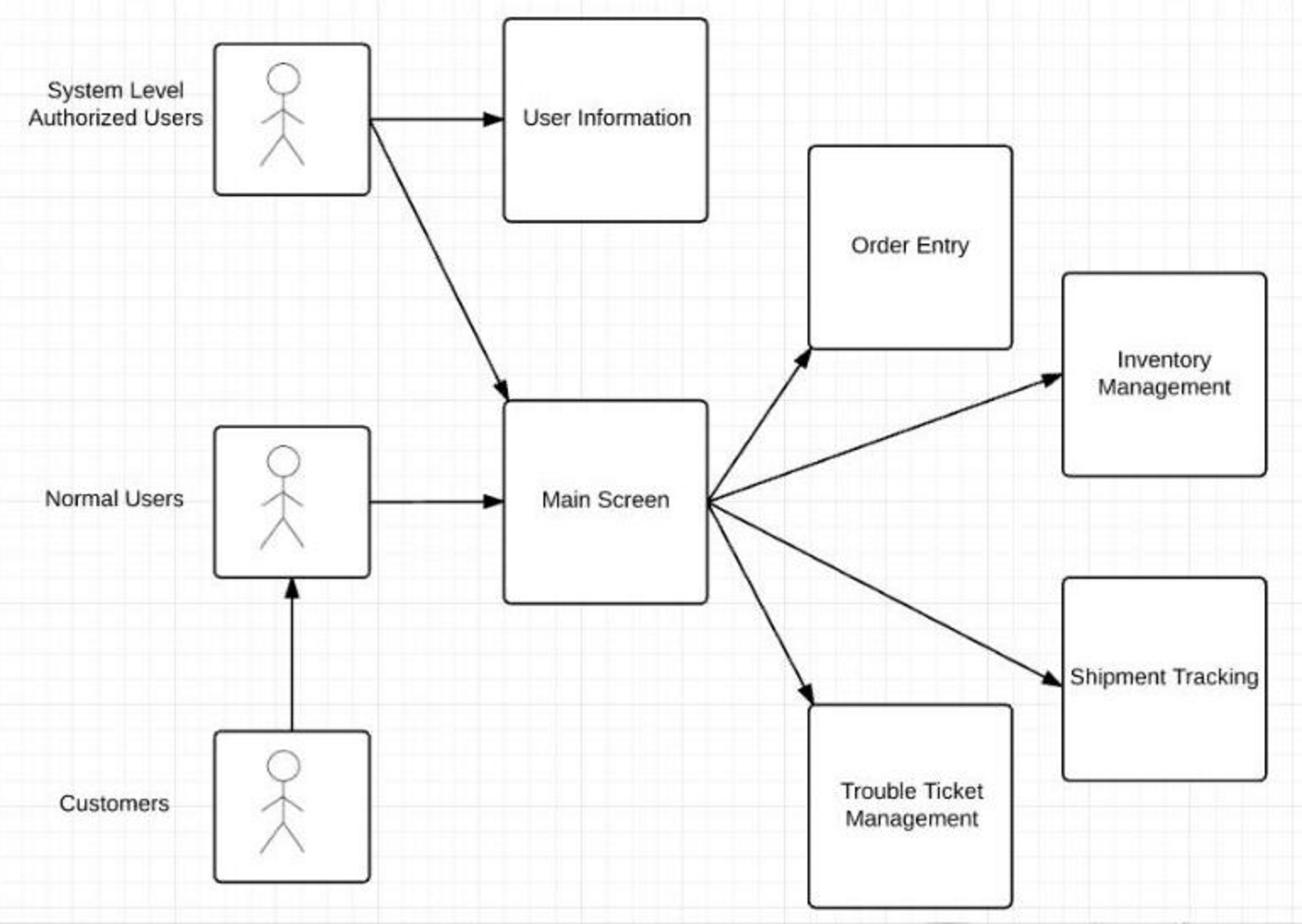
i. Inventory database implementation

ii. Trouble ticket database implementation

iii. Shipment tracking database implementation

iv. Implementation of object based security for login password

v. M-V-C implementation to connect front end with back end

2. So far the only changes we have negotiated with the user is changing the interface from a desktop application to a web application. The requirements suggested using multiple windows to handle different screens and we suggested to the user that we use either multiple tabs and/or windows to handle various screens open at one time.
4. 

/\*\*\*\*\*\*\*\*\*\*\*\*\*\* Informal Notes related to requirements analysis activities \*\*\*\*\*\*\*\*\*\*\*\*\*\*

Requirements Analysis and Project Plan

LINK TO THE POWERPOINT/VIDEO CONTENT:

<https://docs.google.com/document/d/13Z3p7_XzyOMIJFQu42cWkNhjfLVifxKxWKMHYiFiyig/edit?usp=sharing>

2.

a. Members and Jobs - Eric

Dustin - Project Manager, Debugging/ Testing

Eric - Back End

Colan - Front End, Architecture

Anass - Documentation, Back End

Mohammed - Debugging/ Testing, Documentation

Ali - Front End, Architecture, Debugging/ Testing

b. - Eric Development Environment

i. Windows

ii. Ruby on Rails, HTML

iii. SQLite

iv. We decided to use Windows because this is what our developers have experience using. As for the implementation language, we chose Ruby on Rails, both because this is what we have experience using and, to our knowledge, Ruby on Rails is a well rounded, high level language, which makes developing front and back end much simpler than with lower level languages. Ruby on Rails also has lots of resources online which will help make this project more manageable for the time given. Finally, we chose to use MySQL as our Database management system because it’s what we have experience using and we will familiarize ourselves with Ruby on Rails libraries in order to connect with MySQL databases.

c. - Ali

We chose to use Architected Agile because the *Ice Track* system is a perfect candidate for this Common Case. The system is not extremely critical, since any errors in the systems would not cause major harm to anybody; the complexity of the system is rather low, mainly consisting of database lookups and simple

GUI controls. The development environment will use no COTS (Ruby on Rails, SQLite). Key aspects of this Common Case include evolutionary development, welcoming and quickly responding to changes in requirements, close cooperation between client and development team, minimal bureaucracy, and most importantly frequent communication among team members.

d. - Ali

We chose a combination of SCRUM and XP. Having short development cycles, flexibility with requirements, and a system that is not life-critical allows us to take advantage of these methodologies. Given the small size of the group as well the relative simplicity of the project, these methodology are a perfect fit for this system. An important factor the development team must keep in consideration is that frequent communication is key to these methodologies during the development phase. These processes also suggest that every developer has satisfactory coding ability.

e. - Dustin

i. Using Object Points we have deduced that the amount of time that our project requires is 2 person-months

f. - Dustin

Schedule: Spreadsheet

i. Design inventory management system everything else is reliant on

ii. Start: 23 February, Duration:2 weeks

iii. Inventory management

iv. Begining of first subsystem

g. -Dustin Calculated Person-Hours: 1456

Requirements

a. - Colan

i. The *Ice Track* system is designed to allow the admins and users of the software to: place new orders and modify existing ice cream orders; track inventory that’s on hand (both available and pending for shipping), shipped, defective or spoiled which will be stored in a database; track shipping status of items and allow the user to query the status of a shipment according specific attributes; manage trouble tickets stemming from both internal and customer facing problems.

The entire system will run on a menu driven graphical user interface which allows the user to easily navigate between tasks. All processes, transactions, and queries will be executed in real time creating a seamless experience. The system will be built for use primarily with Windows 10 operating system but will also be compatible with previous Windows versions.

ii. Requires Windows 8.1 or Windows 10 with Google Chrome or Firefox installed as these are our testing platforms. A machine with internet access is also required. 4GB of RAM and Duo Core processor recommended.

iii.

b. - Anass & Mohammed

3.

iii. - Mohammed

We have specified 46 features that will be required to make the Ice Track system meets the customer demands. We have not added or deleted any of the original requirements so far. We will be moving to the next stage of development with the current amount of requirements. However, we will evaluate the schedule further and determine if the number needs to be reduced to stay within the time and budget limits without any complications .

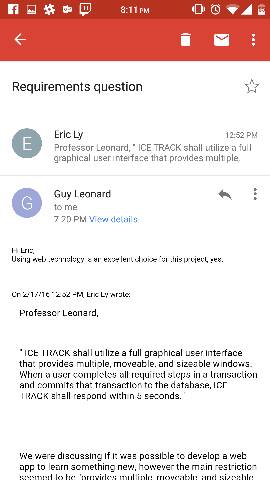
Risk status - Everyone

* New software development tools and frameworks to learn (Ruby on Rails, SQLite)
* New project planning software to learn (OpenProj)
* Time management during group meetings (Not starting working until an hour into meeting)
* Unclear Objectives (What is due, when?)
* Continually changing requirements (Required for the project)
* Inadequately trained team members (Pretty much everyone since we’ve never used Ruby on Rails before)
* Time constraints (15 weeks for 6 students to create an entire system)
* Unclear terminology (eg. ‘3rd level task’)
* Difficult to estimate and schedule.

/\*\*\*\*\*\*\*\*\*\*\* End of informal notes \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



/\*\*\*\*\*\*\*\*\*\*\*\* Informal notes from customer to development team \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



/ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of informal notes from customer \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*