A Pizza Ordering System

TEAM 9002

Project Members:
Domonique Jackson-Russell
Gretchen fahner
Project Management

Domonique Jackson-Russell: Project Manager, Software Quality Assurance Lead
Gretchen Fahner: Software Tester

Summary of Changes
Functional Requirements - Use cases
Concept Definitions
User interface design and implementation
Table of Contents

PROJECT PLAN ......................................................................................................................... 5

Product Scope ............................................................................................................................ 5
1. Order System ......................................................................................................................... 5
   a. E.1 ............................................................................................................................... 5
   b. E.2 ............................................................................................................................... 5
   c. E.3 ............................................................................................................................... 5
   d. E.4 ............................................................................................................................... 5
   e. E.5 ............................................................................................................................... 6
   f. E.6 ............................................................................................................................... 6
2. Delivery System ..................................................................................................................... 6
   a. E.7 ............................................................................................................................... 6

Team Organization ................................................................................................................... 6

Data Management Plan ............................................................................................................ 11
1. Roles and Responsibilities ................................................................................................... 11
   a. Staff Organization ....................................................................................................... 11
2. Project Plan Checks ............................................................................................................ 11
3. Decisions ............................................................................................................................. 11
4. Types of Data ..................................................................................................................... 11
5. The Data Processed ............................................................................................................ 11
6. Policies for Access and Sharing .......................................................................................... 11
7. Resources Needed .............................................................................................................. 11
8. Data Availability ................................................................................................................ 12
9. Data Storage Preservation .................................................................................................. 12
10. Procedures for long-term data storage ............................................................................ 12
11. Preservation of long-term data storage .......................................................................... 12

Project Scope ........................................................................................................................ 13
1. Schedule ............................................................................................................................... 13
   a. Structure ...................................................................................................................... 13
   b. Milestones .................................................................................................................... 15
   c. Deliverables .................................................................................................................. 15
   d. Gantt Chart ................................................................................................................... 17
2. Technical Description ......................................................................................................... 21
   a. Introduction .................................................................................................................. 21
   b. Background .................................................................................................................. 21
   c. User Characteristics .................................................................................................... 21
   d. Data Characteristics ..................................................................................................... 21
   e. Description ................................................................................................................... 21
3. Test Plan .............................................................................................................................. 23
   a. Introduction .................................................................................................................. 23
   b. Test Items .................................................................................................................... 23
   c. Features to be tested .................................................................................................... 23
   d. Features not to be tested .............................................................................................. 23
   e. Approach ..................................................................................................................... 24
   f. Test deliverables .......................................................................................................... 24
Product Scope

- Scope, Team Organization, and Data Management
Product scope

1. Order System Part I
   a. E1. Make a database that can store information it should have:
      i. Master record
         1. Name
         2. Address
      ii. Phone, and
         1. Type of charge account
      iii. The master record will be linked by
         1. Phone number
      iv. A locating mechanism to locate the customer address like:
         1. Subdivision name,
         2. Closest major intersection
   b. E.2 Make customer database that records:
      i. Allow database to receive input from customers and either
         1. Store information if new customer, or
         2. Retrieve stored customer information
      ii. The record should include:
         1. Address and
         2. Delivery information (requirements above)
   c. E.3 take orders and be able to:
      i. Print out receipt
      ii. Receipt should include
         1. A place to sign name if * credit card,
         2. Customer information,
         3. List of items ordered,
         4. Type of order,
            a. Delivery or
            b. Pickup
         5. Amount due
   d. E.4 calculate bills,
   e. E.5. Process payments (check, cash, or credit cards) and
      i. Track type and
      ii. Amount.
   f. E.6 and contain the restaurant menu which should contain:
      i. Full menu,
      ii. Customer GUI access,
      iii. Customer capability to select choices from menu
      iv. A complete list of all items in restaurant

2. Delivery System Part II
   i. E.6 A database that can record pertinent information for locating the customer
address for delivery drivers:
1. Subdivision name,
2. Closest major intersection
3. Landmark
Data Management Plan

1. Roles and Responsibilities
   a. Staff organization.
      i. Three-person team of Software Engineers.
      ii. Domonique Jackson-Russell is Project Manager tasked with managing the constraints of the project, creating obtainable objectives, and ensuring the essential key issues are addressed.
      iii. Gretchen Fahner is the Software Tester tasked with investigating the program by running the application with intent of finding defects and makes sure it functions within all requirements.
      iv. Domonique Jackson-Russell is the Software Quality Assurance Lead tasked with validating that each deliverable meets the specified requirements and the product milestones are done within the specified timeline.

2. Project plan checks
   i. Triple check: The project will meet all requirements maintaining quality as the number one goal by having each team member sign off on each milestone. The triple check will make sure that quality is maintained throughout the process.

3. Decisions
   i. Domonique Jackson-Russell will make all final decisions through the duration of the project related to data issues.

4. Types of Data
   i. The program data will be creating and capturing:
      1. Pizza sizes, flavors, and anything else about pizza
      2. Customer’s information
      3. Address location system
      4. Method of payments
      5. Memory to store customers information

5. The data processed
   i. The data will be processed using sorting that will create categories, which can be call upon.

6. Policies for access and sharing and appropriate protection and privacy
   i. The program will require the user to login before he can add any order or check any information.
   ii. The program will record the user and his transaction.
   iii. The program will record each user searching for a customer.
   iv. Customer data will be encrypted
   v. Having one databases utilizing metadata, which would include customer information with a history of his orders and user information with transactions will allow us to control better on which user is able to see what data.

7. Resources needed to access or use the data
To access the program you need to have a username and password.

The administrator will have the authority to create the credentials for each user.

You need to know the encryption type and key to read the saved data in the database.

8. When will you make the data available?
   i. The data will be available 04/13/2015 with a working prototype. The data will be able to be accessed through a website html. Once feedback is provided and the data is up to standards of the user the final data accessibility will be granted permanently on 04/17/2015. The end user will have all intellectual property rights to the data and full access.

9. Data storage preservation
   a. Archive, repository, and database
      i. We will be using two separate databases, one for customer information with a history of customer orders, and one for user information’s with transactions. This will allow us to control better on which user is able to see what data or access, which database. ex. a regular employee will be able to add/update a customer’s information so he will access the customer information database however he can not create/delete users and will not have access to that database too. An administrator will have the access to add/delete users. A branch owner will have access to add users if needed and not removing them etc..

10. Procedures for intended long-term data storage
    i. A backup database server should be installed so we can have a copy for the entire database and its tables.

    ii. It would be also a nice idea to have a database recording every detail of every transaction a user makes so you can have a record if any issues later on.

11. Preservation of data for the long-term will be:
    i. All data that will help identify the customer will be preserved for a long-term duration, such as:
        1. Names, addresses, payment information, and special identification to locate addresses for delivery.
A. Project scope
   ○ Schedule
      i. Work timeline
      ii. PART I
         a. Start: 02/03/2015
         b. Stop: 02/08/2015
            i. Scope, Team Organization, and Data Management Plan.
      • PART I SECOND
         a. Start: 02/03/2015
         b. Stop: 02/15/2015
            i. Schedule, technical description, and test plan.
      • PART I
         a. Complete 02/22/2015
            i. Scope, Team Organization, and Data Management Plan.
            ii. Schedule, technical description, and test plan.
      • E.1
         a. Start: 02/10/2015
         b. Stop: 02/20/2015
            i. Information database
      • E.2
         a. Start: 02/20/2015
         b. Stop: 02/23/2015
            i. Make customer database that records
      • E.3
         a. Start: 02/23/2015
         b. Stop: 02/24/2015
            i. Take orders
      • E.4
         a. Start: 02/24/2015
         b. Stop: 02/27/2015
            i. Calculate bills,
      • E.5
         a. Start: 02/26/2015
         b. Stop: 02/27/2015
            i. Process payments
      • E.6
         a. Start: 02/27/2015
         b. Stop: 02/28/2015
            i. The restaurant menu
• Requirement I
  a. Start: 03/01/2015
  b. Stop: 03/02/2015
  c. software requirements – a text document (created in PolyRM) and the paper prototype.
     i. Requirements definition document
        1. English text in PolyRM
        2. Paper prototype (drawn by hand or via a drawing tool like Visio or Word).

• Requirement II
  a. Start: 03/02/2015
  b. Stop: 03/09/2015
     i. Requirements specification document
        1. Use case diagram (created by a drawing tool or Dia)
        2. Use case flow of events document (text document, created in Word)
        3. Class diagrams (created in Dia)

• Requirement III
  a. Start: 03/09/2015
  b. Stop: 03/15/2015
     i. Requirements specification document
        1. Class diagrams (created in Dia)
        2. Class documentation (created in Word)
        3. Entity relationship diagram(s) (created by a drawing tool or DB tool)
        4. At least one of each of the following (for documenting classes) EITHER decision table(s) or state transition diagram(s)

• System Design Documents 03/16, 03/30
  a. Conceptual system design:
     i. Start: 02/28/2015
     ii. Stop: 03/16/2015
        1. Report formats (created in Word)
        2. Screen layouts/ shots (screen captures from your prototype application embedded in a text document guiding the user through the application, with the
requirements and use cases being addressed by each screen clearly identified on the bottom of the page – underneath the screenshot

b. Technical design: (high level)
i. Start 02/28/2015
ii. Stop: 03/30/2015
   1. Detailed class diagrams with all methods, attributes and relationships identified (created in Dia)
   2. Supporting text specification (created in Word)
   3. Database table descriptions

- Working Prototype
  a. Start: 02/28/2015
  b. Stop: 04/13/2015

- Final Notebook
  a. Start: 02/28/2015
  b. Stop: 04/13/2015

- Project complete
  a. Start: 02/03/2015
  b. Stop: 04/17/2015

iii. Milestones
    - E.1
    - E.2
    - E.3
    - E.5
    - Requirement I
    - Requirement II
    - Requirement III
    - System Design Documents
    - Working prototype
    - Project complete

iv. Deliverables
    - PART I
      a. 02/15/2015
    - E.6
      a. 03/01/2015
    - Requirement I
- Requirement II
  a. 03/09/2015
- Requirement III
  a. 03/15/2015
- System Design Documents
  a. 03/16/2015
  b. 03/30/2015
- Working Prototype
  a. 04/13/2015
- Final Notebook
  a. 04/13/2015
- Project complete
  a. 04/17/2015
<table>
<thead>
<tr>
<th>v. CHART</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter your deadline as start and end date:</td>
</tr>
<tr>
<td>ART I</td>
</tr>
<tr>
<td>scope</td>
</tr>
<tr>
<td>Team org</td>
</tr>
<tr>
<td>Data Management Plan</td>
</tr>
<tr>
<td>ART I OF II</td>
</tr>
<tr>
<td>schedule</td>
</tr>
<tr>
<td>technical description</td>
</tr>
<tr>
<td>test plan</td>
</tr>
<tr>
<td>1-E.6</td>
</tr>
<tr>
<td>E.1 information database</td>
</tr>
<tr>
<td>E.1 Milestone Complete</td>
</tr>
<tr>
<td>E.2 information database</td>
</tr>
<tr>
<td>E.2 Milestone</td>
</tr>
<tr>
<td>E.3 take orders</td>
</tr>
<tr>
<td>E.3 Milestone</td>
</tr>
<tr>
<td>E.4 calculate bills</td>
</tr>
<tr>
<td>E.4 Milestone</td>
</tr>
<tr>
<td>E.5 process payments</td>
</tr>
<tr>
<td>E.5 Milestone</td>
</tr>
<tr>
<td>E.6 the restaurant menu</td>
</tr>
<tr>
<td>E.6 Milestone</td>
</tr>
<tr>
<td>REQUIREMENT I-III</td>
</tr>
<tr>
<td>requirement I</td>
</tr>
<tr>
<td>requirement I Milestone</td>
</tr>
<tr>
<td>requirement II</td>
</tr>
<tr>
<td>Task Name</td>
</tr>
<tr>
<td>---------------------------</td>
</tr>
<tr>
<td>requirement II Milestone</td>
</tr>
<tr>
<td>requirement III</td>
</tr>
<tr>
<td>requirement III Milestone</td>
</tr>
<tr>
<td>system design</td>
</tr>
<tr>
<td>system design Milestone</td>
</tr>
<tr>
<td>conceptual system design</td>
</tr>
<tr>
<td>conceptual system design Milestone</td>
</tr>
<tr>
<td>technical design</td>
</tr>
<tr>
<td>technical design Milestone</td>
</tr>
<tr>
<td>working prototype</td>
</tr>
<tr>
<td>working prototype Milestone</td>
</tr>
<tr>
<td>final notebook</td>
</tr>
<tr>
<td>final notebook Milestone</td>
</tr>
<tr>
<td>project complete</td>
</tr>
<tr>
<td>project complete Milestone</td>
</tr>
<tr>
<td>Mar 1</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>F</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Enter your deadline as start and end date:

- E.1-E.6

- E.4: Calculate bills
  - E.4 Milestone
- E.5: Process payments
  - E.5 Milestone
- E.6: Restaurant menu
  - E.6 Milestone

- Requirement I
- Requirement I Milestone
- Requirement II

16
<table>
<thead>
<tr>
<th>Days</th>
<th>Feb 1</th>
<th>Feb 8</th>
<th>Feb 15</th>
<th>Feb 22</th>
</tr>
</thead>
<tbody>
<tr>
<td>S M T W T F</td>
<td>S M T W T F</td>
<td>S M T W T F</td>
<td>S M T W T F</td>
<td></td>
</tr>
<tr>
<td>PART I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>scope</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team org</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Management Plan</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PART I OF II</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>schedule</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>technical description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>test plan</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E.1 information database</td>
<td>E.1 Milestone Complete</td>
<td>E.2 information database</td>
<td>E.2 Milestone</td>
<td></td>
</tr>
<tr>
<td>E.3 take o</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E.3 Milesto</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
VII. Technical Description

- Introduction
  - The pizza ordering system 1.0 is a program that will be capable of inputting data and retrieving data to support the ordering process. The program will have all necessary flavors of crusts, crust types, and pizza ingredients accessible by all customers. The program will be actively retrieving information and dispatching it to the main server of the pizza establishment to queue orders in time stamped chronological order.

- Background
  - The technical plan will cover the user characteristics, data characteristics, and requirements to run program.

- User characteristics
  - The customer will have to be capable of understanding how to input data by simply clicking in the areas of the interface that indicates input data. The customer will also have to understand how to navigate the pizza ingredient section to make the correct choices of the desired product. The easy interface of the program will facilitate the customer to understand how to select each pizza ingredient.
  - The pizza employees will have to have a working knowledge of how to input data into the pizza database. The employee will need to have a PC OS, Mac OS, Mobile Internet, android app, or iPhone app with Internet connection.

- Data characteristics
  - The interface will provide a front screen called “set my address to order” this will allow customer to input their information to be stored in the pizza programs database and a button to clicked when finished.
  - The program will need a server to store the data information to be saved for the purpose of customer identification using the master key as the customer’s number.
  - The menu page will have most popular pizzas for a quick selection.
  - The menu page will have “customize your own pizza” which will include crust sizes, crust flavors, and ingredient to go on top of pizza.
  - The interface will have a running tally of order calculation on the right hand screen to keep customer informed of current bill.
  - The interface will have a button to click when order is finished that will take the customer to the order page to retrieve payment type from customer.
  - The customer will have the opportunity to save payment history in the program database for future quick process.

- Description
  - A program that can run on any windows OS, Mac OS, mobile Internet, android app, or iPhone app with Internet connection. The interface will consist of an interactive click to make selections of items. (See figure 1 illustration).
*All rights reserved to Papa Johns Pizza TM is used for educational purposes.
VIII. Test Plan

Test plan identifier: Pizz11895

Contents

Introduction
Test items
Features to be tested
Features not to be tested
Approach
Test deliverables
Testing tasks
Responsibilities
Approvals

1. Introduction
The pizza ordering system 1.0 is a program that will be capable of inputting data and retrieving data to support the ordering process. The program will have all necessary flavors of crusts, crust types, and pizza ingredients.

2. Test items
Information database

3. Features to be tested
E.1 information database to make sure the master record that includes name, address, and phone number link to the phone number for locating customer.
E.2 test items customer database records
E.3 Order taking
E.4 bill calculation
E.5 process payments
E.6 restaurant menu, and working prototype of complete system

4. Features not to be tested
Requirement I
Requirement II
Requirement III
5. Approach
A Test Strategy document is a high level document and normally developed by project manager. This document defines “Software Testing Approach” to achieve testing objectives. The Test Strategy is normally derived from the Business Requirement Specification document. The Test Strategy document is a static document meaning that it is not updated too often. It sets the standards for testing processes and activities and other documents such as the Test Plan draws its contents from those standards set in the Test Strategy Document.

6. Test deliverables
Please refer to IV.

7. Testing tasks
Triple check: The project will meet all requirements maintaining quality as the number one goal by having each team member sign off on each milestone. The triple check will make sure that quality is maintained throughout the process.

8. Responsibilities
Domonique Jackson-Russell is Project Manager tasked with managing the constraints of the project, creating obtainable objectives, and ensuring the essential key issues are addressed. Gretchen Fahner is the Software Tester tasked with investigating the program by running the application with intent of finding defects and makes sure it functions within all requirements. Domonique Jackson-Russell is the Software Quality Assurance Lead tasked with validating that each deliverable meets the specified requirements and the product milestones are done within the specified timeline.

9. Approvals
Domonique Jackson-Russell will make all final decisions through the duration of the project related to data issues.
I. Requirement I
   a. Definition
      i. Technical definitions
         1. Data: The file where the menu items, customer information, customer past orders, and new orders are stored.
         2. Order queue: a list of customer orders place first-in, first-out.
         3. Graphical User Interfaces (GUI)
      ii. Non-technical definitions
         1. Customer: The person that orders items from restaurant.
         2. Cashier: The person that takes the customers orders.
         3. Manager: The person in charge of employees.
         4. Items: Any inventory for purchase on menu.
   b. Customer interaction
      1. The pizza program 1.0 is the typical ordering and delivery retrieval system. How the program would work in typical conditions is a customer or employee would access the database and enter the required information. The goal of the program is to develop a data collection that would allow new customers to input their personal data (name, address, number, payment type) and track personal preferences of payment type and food selections. The program would also retrieve old customers information to cut down on ordering time. The program interface will be visually attractive, with effective navigation tools that will make it easy to maneuver without any confusing on how to navigate through. Depending on if the customer is an old or new customer will determine the setup process. If it is a new customer the customer will have to fill out their name, address, phone number in the beginning and have the option to save their account or proceed as a guest. Once the new customer fills the indicated areas out they will have to chose if it is for delivery or takeout and then be able to proceed to the menu to order. If the customer is an old customer that allowed the database to store their information the customers phone number will retrieve the master record. Once the customer enters their number, the customers address, name, number, and payment preference will be retrieved and auto fill the information. The old customer will only have to click the delivery or takeout option to proceed. The customers will then be allowed to proceed to the full menu of the pizza restaurant. The menu page will be very simple to cut down on confusion with different pizza sizes, crust options, ingredients for quarter, half, or whole pizza, and any special instructions. The customer selects their desired pizza and on the right hand of the page will have a live up to date bill calculating all prices selected. If the customer is satisfied, they will have the option below the bill to proceed to checkout. If the
customer is not satisfied, the customer will be allowed to erase all selections and start over. Once they hit checkout the customer will be brought to another page that shows the final cost breakdown. The customer will be then given the option to select payment type in the form of credit card, debit card, cash payment. The customer will also be able to include a tip if it is a delivery and that will be calculated in the bill. If the customer is a new customer they will be allowed to enter the payment type and if it is an electronic payment will be given the options to save the payment type for future reference. The customer will also be able to enter a different address if the delivery is not to the address linked to the customers account. After the customer has filled in the required information, they will see a button at the bottom finish order. Once they click finish order they will be brought to another page that shows the customer a detailed report of their information, the order details, the receipt breakdown, the payment type, and estimated time for pizza to be done. If the customer’s payment information is not valid, they will receive a message indicating the payment did not go through. Once payment is approved, the data will be transmitted to the main server at the pizza restaurant. Once delivered the customer would receive a physical receipt.

ii. **Employee interaction**

1. Depending on how the pizza is ordered will determine the way the program will interact with the server. If the customer does the order, the electronic order will be automatically sent to the pizza restaurants screen, which will be highlighted bright red. The screen will have split screens showing two types of orders one for delivery and one for takeout. All the ordering information in the database will be sorted in a queue by time so that the orders can be made from earliest order to latest order. The electronic orders will have the capability of being overridden and edited by any employee. The program will have an edit button on the screen so that they can edit orders. Once the employee hits edit they will have the option of deleting or adding items, changing payment type and information, changing addresses, and canceling the order all together. The employees will also be able to make comments to the customer’s stored information such as, how to locate the customers address easier using landmark indications. The system will also have a GPS address tracker which will be a tab that can be clicked on which will bring the employee out of the split screen to the tracker screen that will facilitate in clustering orders by addresses closeness in groups. Providing this will allow the delivery drivers to be more efficient in delivering pizzas to all the same area instead of going back and forth. If the employee has to add the customers order in the screen will be the same setup as the customers mentioned above. The only difference is the option to edit
and add landmark information to the customers file. In addition, the adding function of a cash register that the employee must manually input the data of cash in and the system will calculate cash out.

2. Management

The management will have a special database to access to be able to customize the menu at anytime. Once the manager enters the editing area they will be able to easily update menu with words and images. Also, change any information. The changes will become effective immediately once the submission occurs.

<table>
<thead>
<tr>
<th>Version #</th>
<th>Implemented By</th>
<th>Revision Date</th>
<th>Approved By</th>
<th>Approval Date</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1.0]</td>
<td>[Domonique Jackson-Russell, Gretchen Fahner]</td>
<td>[02/25/2015]</td>
<td>[Domonique Jackson-Russell]</td>
<td>[02/25/2015]</td>
<td>[mk]</td>
</tr>
</tbody>
</table>

1. Functional Requirements

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQ-1</td>
<td>The customer will sign in, and order on the computer, which will allow the customer to enter personal information to be saved in data on the server or continue as guest.</td>
</tr>
<tr>
<td>REQ-2</td>
<td>The customer on the computer will be allowed to have access to all the items on the menu.</td>
</tr>
<tr>
<td>REQ-3</td>
<td>The customer on the computer will be allowed to choose all items by clicking the desired items with option of customizing order.</td>
</tr>
<tr>
<td>REQ-4</td>
<td>The customer on a computer will be able to check order type (delivery or pickup).</td>
</tr>
<tr>
<td>REQ-5</td>
<td>The customer on the computer will be able to keep track of pizza cost, see total currency to be paid, be given payment options, process payment order, and be given an electronic receipt.</td>
</tr>
<tr>
<td>REQ-6</td>
<td>The customer will on the computer will be able to submit order and be placed</td>
</tr>
</tbody>
</table>
into a queue, which keeps track of what customer ordered first.

REQ-7  The customer will also be able to use the phone or walk-in to order with the cashier who will use a computer.

REQ-8  The cashier computer will allow the full menu to be viewed.

REQ-9  The cashier computer will be able to place orders, which include customer information, customer order type, any special instructions, GPS landmark, and payment type per customer.

REQ-10 The cashier computer will allow the cashier to see record of time each customer has been waiting for order.

REQ-11 The cashier computer will be able to print out receipt or send electronic receipt.

REQ-12 The cashier computer will see when an order is done and ready for delivery or pickup.

REQ-13 The cashier computer will keep an up to date order system that deletes orders that have been delivered or picked up.

REQ-14 The cashier computer will be able to add, edit, or remove an order.

REQ-15 The cashier computer will be able to see the GPS of address order clusters for delivery efficiency purposes.

REQ-16 The manager computer will be able to add, edit, or remove menu items.

REQ-17 The manager computer will be able to add, edit, or remove menu items.

2. Non-Functional Requirements

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQ-18</td>
<td>The program must be aesthetically pleasing and correlate with the pizza restaurant.</td>
</tr>
<tr>
<td>REQ-19</td>
<td>Security must be high priority to ensure non-users cannot access restaurant or customer sensitive information.</td>
</tr>
<tr>
<td>Identifier</td>
<td>Requirements</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
</tr>
<tr>
<td>REQ-20</td>
<td>The manager should make sure that there is an orientation class on proper use of program.</td>
</tr>
<tr>
<td>REQ-21</td>
<td>The server should have a back up in case of failures to avoid data loss.</td>
</tr>
<tr>
<td>REQ-22</td>
<td>The program should have a low mean time between failures.</td>
</tr>
<tr>
<td>REQ-23</td>
<td>The system must work on any type of device.</td>
</tr>
<tr>
<td>REQ-24</td>
<td>The LCD monitors should be an efficient size to ensure full maximization of tabs.</td>
</tr>
<tr>
<td>REQ-25</td>
<td>The system should work on any PC, IOS, Android, or IOS device.</td>
</tr>
<tr>
<td>REQ-26</td>
<td>The system should be easy to use for all technical levels.</td>
</tr>
<tr>
<td>REQ-27</td>
<td>The system should be easy to trouble shoot.</td>
</tr>
<tr>
<td>REQ-28</td>
<td>The system should work on any PC, IOS, Android, or IOS device.</td>
</tr>
<tr>
<td>REQ-29</td>
<td>The system should work on any PC, IOS, Android, or IOS device.</td>
</tr>
</tbody>
</table>

3. **Graphical User Interface Requirements**

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQ-30</td>
<td>The system should show existing customer or new customer screen.</td>
</tr>
<tr>
<td>REQ-31</td>
<td>The system should show input for customer data during initial sign-in.</td>
</tr>
<tr>
<td>REQ-32</td>
<td>The system should display menu items with prices.</td>
</tr>
<tr>
<td>REQ-33</td>
<td>The system should show customization option.</td>
</tr>
<tr>
<td>REQ-34</td>
<td>The system should show order total.</td>
</tr>
<tr>
<td>REQ-35</td>
<td>The system show order type (delivery or pickup).</td>
</tr>
<tr>
<td>REQ-36</td>
<td>The system should show input payment information.</td>
</tr>
<tr>
<td>REQ-37</td>
<td>The system should show receipt of processed order.</td>
</tr>
<tr>
<td>REQ-38</td>
<td>The system should show customer order queue.</td>
</tr>
<tr>
<td>REQ-39</td>
<td>The system should show main homepage.</td>
</tr>
<tr>
<td>REQ-40</td>
<td>The system should show customer information, order, and queue in one tab.</td>
</tr>
<tr>
<td>REQ-41</td>
<td>The system should show order edit option.</td>
</tr>
<tr>
<td>REQ-42</td>
<td>The system should show complete menu.</td>
</tr>
<tr>
<td>REQ-43</td>
<td>The system should show delivery screen with customer address and GPS.</td>
</tr>
<tr>
<td>REQ-44</td>
<td>The system should show edit menu.</td>
</tr>
</tbody>
</table>

c. Preliminary Paper Prototype Screen Layout
   i. Customer paper prototype
d. Preliminary Paper Prototype Screen Layout
   i. Employee paper prototype
I. Requirement Part II

a. Use-Case Diagram(s) / Class Diagram
Flow of Events

X Flow of Events for the <name> Use case
X 1. Preconditions
X 2. Main Flow
X 3. Sub-flows (if applicable)
X 4. Alternative Flows

1.0 Flow of Events for the Customer Online Order with Delivery Use Case
1.1 Preconditions
The website must be completely operational before the Use case can begin. The payment system must be able to communicate with required authentication resources.

1.2 Main Flow
This use case begins when the customer links to the Pizza ordering website. A menu of available items is displayed with options for adding each main item to the cart. After a main item has been selected, an options menu is displayed for crust type, toppings, etc. The system then prompts for CHECKOUT? If the activity is yes, Checkout, the user is again prompted for Registered User information, or New User Information. For New Users, a page to enter all delivery and payment information is displayed. For Registered users, a Login Page, followed by a payment and delivery Information confirmation is displayed. When the driver is out for delivery, a notification will be sent to the user with the phone number indicated in the customer information.

1.3 Sub-flows
S-1: Add a User-created item. The system will allow the user to create a pizza “from scratch” starting with just a selection of crust type, sauce type, and cheese type. Toppings can then be added to either one side or the other, or the whole pizza.

S-2: Change of User Information
Upon being prompt to confirm User information, the user may select the option to update Address, Phone Number, etc. After updating, the user can save the information as a new address, and updated address and set the default for future orders.

1.4 Alternative Flows
E-1: An invalid Username and password is entered. The user will be prompted to create a new profile or use their unique key to retrieve information.
E-2: An invalid address is entered. Addresses will be checked for standard formatting (including phone numbers) to match a set input. User will be warned and prompted to re-enter or offer to transfer the information to a Pick-Up order.

2.0 Flow of Events for the Customer Call in Order for Pick-Up

2.1 Preconditions, the terminals at the pizza location must be operational.

2.2 Main Flow
This use case begins with a user calling in to make an order. The number used to call if the ID was available will reference customers. If available, the most-previous order will be displayed on the screen with the option to duplicate. If not available, the operator will ask for (at minimum) a Customer telephone number.
The terminal will then display the store menu version. The customer will place an order by verbally describing the items desired. The operator will need to be familiar with the named-item shortcuts that are commonly ordered items. For New Users, a request for payment information will be displayed. This can be skipped with the PAY AT LOCATION option. For Registered users, previous payment option will be displayed.

2.3 Subflows
   S-1: Add a User
       When adding a new user with (at minimum) telephone information, a prompt will be created to enter the users name.
   S-2: Change of User Information
       If the current information is no longer up to date, the option to update Address, Phone Number, etc will be displayed.

2.4 Alternative Flows
   E-1: An invalid phone number is entered. Numbers will be checked for standard formatting to match a set input.

3.0 Flow of events for the Edit Menu Use Case

3.1 Preconditions
   The employee modifying the current menu options must be an administrator or given special privileges (based on a 2-hour timeout) to change and update/edit menu items.

3.2 Main Flow
   In an elevated privilege mode, all prices, icons and names are able to be edited. Selecting any field in this mode will bring up all available attributes. Manager approval must be given (by re-entering password) any time a price is change or a menu item is added or deleted.

3.3 Sub-flows

3.3.1 Add a New Item
       Adding a new item to the menu will request a price, name, and icon input field to be filled out. The icon field is auto-populated with the name of the item, and can be edited.

3.3.2 Delete Item
       A long-click (holding the icon/name) of any item will produce a new option for deleting the menu item.

3.4 Alternative Flows
   E-1: If an invalid price for a menu item is entered (within a percent of the item cost stored in managerial database), a prompt will ask for a new price to be entered.
E-2: If a name is entered that is already in use, an error will be displayed with a prompt to enter a new name.
I. Requirement Part III

a. ERD

b. Class Specification Document

i.

Class Name: Customers Account
Category: data
External Documents:
Export control: Do Not Release
Hierarchy:
Superclasses: Data
Associations:
Operation Name: n/a
Public member of: Customers Account
Documentation:
// stores customer data
Preconditions:
// for all orders
Name is not NULL;
Phone Number is not NULL;
// for delivery orders
Street Address is not NULL;
City is not NULL;
Zip Code is not NULL; // can be calculated by previous information
-- Apt codes / delivery notes are also stored here
Private Attributes: none
Public Attributes: Name; Street Address; City; State; Zip Code; Phone Number

Class Name: Purchase Order
Category: data
External Documents:
Export control: Do Not Release
Hierarchy:
Superclasses: Data
Associations:
Operation Name: n/a
Public member of: Purchase Order
Documentation:
// stores order data
Preconditions:
// for all orders
Name is not NULL;
Phone Number is not NULL;
// for delivery orders
Customer Address is not NULL;
Private Attributes: // from Customer Account
Customer Phone Number; Customer Address
Public Attributes: Customer Name; Delivery or Pickup; Payment Type; Item order number; Order time

Class Name: Menu Items
Category: data
External Documents:
Export control: Public
Class Name: Food Items
Category: data
External Documents:
Export control: Public
Hierarchy:
Superclasses: Menu items
Associations:
Operation Name: price()
Public member of: Menu items
Documentation:
// stores specific food item data
Preconditions:
// all cost / price values > 0
Semantics:
price = item_cost * pi_ratio
tax = price * purchase.tax_rate
Private Attributes: price
Public Attributes: Food Item number; Description
c. Decision table
e. Paper Prototype Screen Layout
   i. Customer paper prototype please click on link for preview of interaction:
      1. https://popapp.in/w/projects/54eead5d8c60e0633fb7ace0/preview

   ii. Employee and Management paper prototype please click link for preview of interaction:
      1. https://popapp.in/w/projects/54eeac770a567b463fb45e27/preview
• Main page
  o Click either “existing customer” or “new customer” to proceed.
  o Depending on which button the customer selects it will take them to either “existing customer” page or “new customer” page.
  o On the bottom the customer can click on “About Us,” “Customer Service,” “Nutritional Facts,” and “Contact Us.”
• If “existing customer” is clicked the customer will enter phone number to access the customers saved data then click the “pizza” to go to menu page.
• If a “new customer” the customer will have to fill out the customer information and have the option to save the data for future reference or continue as “guest.” Customer clicks “continue” to go to “main menu.”
• “Main menu” will have the option to click six different options of pizza styles “favorites” which are a list of popular customer choices, “create your own” which allows you to customize your pizza, “specials” any specials the restaurant is having at that time, “vegan,” “gluten free”, and “ready to eat” which is the bare essential pizza already premade.

• Once the customer selects their option it will depending on selection already have the ingredients preselected on the “chose your toppings page.”
• Customer can create a pizza or add to a pre-selected pizza.
• The right hand side shows a total, as selections occur it keeps running tally to keep the customer informed of price.
• The customer can click on "remove" which will erase all selections and take the customer back to "main menu."
• The customer can click “order this pizza” or “checkout” to be sent to the “receipt page.”
• The customer can view order and make any changes on this page.
• Once customer is happy, the customer can click on “checkout.”
• Old customer with saved payment type it will auto fill so they will not have to enter information.
• New customer will not have to enter in address it will auto fill but will have to enter in payment type.
• The customer will then click “complete order” and it will process payment.
The order is complete and sent to the server to be processed by the pizza establishment.
Illustration of Employee and Manager prototype (Fig. 2)

- Homepage “Main menu” that has links to “edit the menu,” “order page,” “GPS page,” and “customer order input.”
- This page will auto select when there is inactivity in the program like a screen saver.
• The “order page” is where the customers order is calculated and shows the total amount and the employee can take payment and print receipt.

• Employee can track incoming orders with the queue having the earliest in on the top of the scroll down screen. It will tell the employee the customers name, address, and order.

• Once the order is done and ready for delivery or pickup, depending on type of order the appropriate box will have a red “!” and when clicked on the same box it will show the customer information.

• Once the order is delivered or picked up the queue will erase the order and the next order in the queue will move in place.

• The employee can click “edit” to edit the customers order.

• The employee can click “GPS” to see the delivery clusters based on incoming orders addresses.

• The employee can click on “menu change” but will have to put in a code to have access to the privileged area.

• The employee can click on “Main” to go to the main menu.
The employee can edit the customer's order, address, or payment from this page.

The employee can cancel an order from this page.

Once done, the employee clicks "main" to get back to the main menu.
• The employee can take orders from this page.
• Once the order is complete, it will transfer to the “order page” under type of order (delivery or pickup) and have total.
• The employee can access other pages by clicking “edit,” “order page,” and “GPS.”
Employees can access the “GPS Page,” which will show the addresses on the left, and the clusters of deliveries on the right to show how close in proximity the orders are.

The GPS cluster will indicate zones for each driver for efficiency purposes.

The employee can access other pages by clicking on “edit,” “main,” and “menu change.”
• The employee can access the "Edit Menu Page" only by properly entering in a code.
• The employee can easily add images in the webpage.
• The employee can easily add quantity and pricing.
• The employee can easily put any specials in the webpage.
• The employee can enter promo codes on the webpage.
II. Technical Design
Class Name: **Customer**
Category: data
External Documents: 
Export control: Do Not Release
Hierarchy:
   Superclasses: Data
Associations:
Operation Name:
   updateName()
   updatePhoneN()
   setState()
Public member of: Customer
Documentation:
   // stores customer name, phone number, and website login information
Preconditions:
   // for all orders
   Name is not NULL;
   Phone Number is not NULL;
Private Attributes: none
Public Attributes: Name, ID; Phone Number

Class Name: **Customer Address**
Category: data
External Documents: 
Export control: Do Not Release
Hierarchy:
   Superclasses: Data, Customer
Associations:
Operation Name:
   updateLine1()
   updateLine2()
   updateCity()
   State() // this is always GA*
   getZip()
   setNotes()
Public member of: Customer, GPS System
Documentation:
   // stores customer address information
Preconditions:
   // for delivery orders
   Street Address is not NULL;
   City is not NULL;
   Zip Code is not NULL; // can be calculated by previous information
   -- Apt codes / delivery notes are also stored here
Private Attributes: none
Public Attributes: Line 1, Line 2; City; State; Zip Code
Class Name: **Purchase / Order**

Category: Data

External Documents:

Export control: Conditional Release

Hierarchy:

Superclasses: Data

Associations:

Operation Name:

  setCustomer()
  isOnline()  // TRUE if web-submitted
  setPayType()
  setOrderNum()  // auto-generated
  applyTime()  // write time of order
  applyDiscount()

Public member of: Purchase Order

Documentation:

  // stores order data

Preconditions:

  // for all orders
  Customer is not NULL;
  Payment Type is not NULL;
  // for Credit Card Type
  Payment Info is not NULL -- not stored in database;

Semantics:

  discount = coupon_code_max * FoodMenu_Item.Price

Private Attributes:

  Customer Phone Number; Customer Address, Payment Info

Public Attributes:

  Customer ID; Delivery; Payment Type; Order number; Order time

---

Class Name: **Employee**

Category: data

External Documents:

Export control: Do Not Release

Hierarchy:

Superclasses: Data

Associations:

Operation Name:

  updateAddress()
  updateCity()
  updateZip()
  updatePhoneN()
  isManager()
  setTaxData()
  setSystemLogin()

Public member of: Employee
Documentation:
    // stores all required Employee information
Preconditions:
    // all fields are Required
Private Attributes: Street Address; City; State; Zip Code; Phone Number; Hire Date
Public Attributes: Name, ID, Manager

Class Name: **Food / Menu Item**
  Category: data
  External Documents:
  Export control: Public
  Hierarchy:
    Superclasses: Data
  Associations:
  Operation Name:
    createItem()
    updatePrice()
    delete()
    updateDesc()
    setType()
  Public member of: Food_Menu Item; Purchase_Order
Documentation:
    // stores all menu data
Preconditions:
Semantics:
    // all price values > 0
    price = item_cost * pi_ratio
Private Attributes:
Public Attributes: Item Number, Order #, Description, Price, Type

Class Name: **Ingredients / Toppings**
  Category: data
  External Documents:
  Export control: Public
  Hierarchy:
    Superclasses: Food_Menu Item
  Associations:
  Operation Name:
    getQuantity()
    updateQuantity()
    updateDiscount()
    ifFresh()    // effects Discount max
  Public member of: Menu items
Documentation:
    // stores specific food item data
Preconditions:
// if getQuantity() == 0, remove from options menu
// max Discount increases based on time food has been available

Semantics:
Private Attributes: Fresh, Shelf Date
Public Attributes: Item #, Quantity, Discount

Class Name: **GPS System**
Category: data
External Documents:
Export control: Do Not Release
Hierarchy:
   Superclasses: Data
Associations:
Operation Name:
   getCustomerData() // pulls Customer Address Information
   updateDisplay()    
   calcEstimate()     
Public member of: Menu items
Documentation:
   // stores address and mapping information for time estimates/navigation
Preconditions:
   // only used for Delivery orders
Semantics:
   // Estimate == drive time + 25 min minimum for Dinnertime deliveries
Private Attributes: Customer ID, Delivery Address
Public Attributes: Store Address, Delivery Estimate

### Database Design

<table>
<thead>
<tr>
<th>Customer</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Label</strong></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>Name</td>
<td>varchar</td>
</tr>
<tr>
<td>ID</td>
<td>int</td>
</tr>
<tr>
<td>Label</td>
<td>Type</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>Customer ID</td>
<td>int</td>
</tr>
<tr>
<td>Line 1</td>
<td>varchar</td>
</tr>
<tr>
<td>Line 2</td>
<td>varchar</td>
</tr>
<tr>
<td>City</td>
<td>varchar</td>
</tr>
<tr>
<td>State</td>
<td>varchar(2) : GA</td>
</tr>
<tr>
<td>Zip Code</td>
<td>int (XXXXX)</td>
</tr>
<tr>
<td>Delivery Notes</td>
<td>varchar (long)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Label</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>varchar</td>
</tr>
<tr>
<td>ID</td>
<td>int</td>
</tr>
<tr>
<td>System Login</td>
<td>varchar (firstinit.lastname)</td>
</tr>
<tr>
<td>Street Address</td>
<td>varchar</td>
</tr>
<tr>
<td>----------------</td>
<td>---------</td>
</tr>
<tr>
<td>City</td>
<td>varchar</td>
</tr>
<tr>
<td>State</td>
<td>varchar(2) : GA</td>
</tr>
<tr>
<td>Zip Code</td>
<td>int (3XXXX)</td>
</tr>
<tr>
<td>Phone Number</td>
<td>varchar (XXX-XXX-XXXXX)</td>
</tr>
<tr>
<td>Manager</td>
<td>tinyint</td>
</tr>
<tr>
<td>Hire Date</td>
<td>datetime</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Purchase / Order</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Label</strong></td>
</tr>
<tr>
<td>Customer ID</td>
</tr>
<tr>
<td>Online Order</td>
</tr>
<tr>
<td>Delivery</td>
</tr>
<tr>
<td>Payment Type</td>
</tr>
<tr>
<td>Order Number</td>
</tr>
<tr>
<td>Order Time of Day</td>
</tr>
<tr>
<td>Payment Info</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>GPS System</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Label</strong></td>
</tr>
<tr>
<td>Store Address</td>
</tr>
<tr>
<td>Customer ID</td>
</tr>
<tr>
<td><strong>Delivery Address</strong></td>
</tr>
<tr>
<td>----------------------</td>
</tr>
<tr>
<td><strong>Delivery Estimate</strong></td>
</tr>
</tbody>
</table>

### Food / Menu Item

<table>
<thead>
<tr>
<th><strong>Label</strong></th>
<th><strong>Type</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Item Number</td>
<td>int</td>
</tr>
<tr>
<td>Purchase / Order #</td>
<td>int</td>
</tr>
<tr>
<td>Description</td>
<td>varchar</td>
</tr>
<tr>
<td>Price</td>
<td>decimal</td>
</tr>
<tr>
<td>Type</td>
<td>varchar</td>
</tr>
</tbody>
</table>

### Ingredients / Toppings

<table>
<thead>
<tr>
<th><strong>Label</strong></th>
<th><strong>Type</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Item ID #</td>
<td>int</td>
</tr>
<tr>
<td>Quantity</td>
<td>int</td>
</tr>
<tr>
<td>Discount</td>
<td>decimal</td>
</tr>
<tr>
<td>Fresh</td>
<td>tinyint</td>
</tr>
<tr>
<td>Shelf Date</td>
<td>timestamp</td>
</tr>
</tbody>
</table>
From  
"Jim Rutherfoord" <jruther3@spsu.view.usg.edu>  

To  
"Jim Rutherfoord" <jruther3@spsu.view.usg.edu>  

Subject  
Software Engineering I XLS Group M6 Spring 2015 CO  

Message  
I looked at your Plan 1. I have one suggestion. You mention having two customer databases for security. Having data in two places can add problems when making sure both are synchronized correctly. The more recent practice seems to be to have data once and restrict access to data fields by using views through the metadata controlled by the users' permissions.

J. Rutherfoord
Weekly Status Reports (reports below)
1. WEEKLY STATUS REPORT

To:
Professor Rutherfoord

From:
Domonique Jackson-Russell
Gretchen Fahner

Subject: status
Period Ending: 02/08/2015
Self Assessment: Green Yellow, Red

<table>
<thead>
<tr>
<th>ACTIVITIES COMPLETED THIS WEEK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed Deliverables:</td>
</tr>
<tr>
<td>➢ PART I</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACTIVITIES IN PROCESS</th>
<th>NEXT ACTION</th>
<th>DUE DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Scope, Team Organization, and Data Management Plan</td>
<td>➢ Finish Part I</td>
<td>➢ 02/08/2015</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACTIVITIES TO BE STARTED NEXT WEEK</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ PART I second half</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LONG TERM PROJECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Requirement I</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISSUES FOR IMMEDIATE ATTENTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Please send your resume!!</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KEY TEAM INTERDEPENDENCIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Gretchen task is Scope (percentage 80%), update (100%)</td>
</tr>
<tr>
<td>➢ Task Team Organization (percentage 100%), and update (100%)</td>
</tr>
<tr>
<td>➢ Domonique Data Management Plan (percentage 70%) update (100%).</td>
</tr>
</tbody>
</table>
# 2. WEEKLY STATUS REPORT

To:  
Professor Rutherfoord  

From:  
Domonique Jackson-Russell  
Gretchen Fahner  

Subject: status  
Period Ending: 02/15/2015  

Self Assessment: Green, Yellow, Red

## ACTIVITIES COMPLETED THIS WEEK

<table>
<thead>
<tr>
<th>Completed Deliverables:</th>
</tr>
</thead>
<tbody>
<tr>
<td>PART I second half</td>
</tr>
</tbody>
</table>

## ACTIVITIES IN PROCESS

<table>
<thead>
<tr>
<th>ACTIVITIES IN PROCESS</th>
<th>NEXT ACTION</th>
<th>DUE DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule, Technical description, and test plan.</td>
<td>Finish Part I (02/22/2015)</td>
<td>02/15/2015</td>
</tr>
</tbody>
</table>

## ACTIVITIES TO BE STARTED NEXT WEEK

<table>
<thead>
<tr>
<th>Requirement I</th>
</tr>
</thead>
</table>

## LONG TERM PROJECTS

<table>
<thead>
<tr>
<th>Requirement i-iii complete</th>
</tr>
</thead>
</table>

## ISSUES FOR IMMEDIATE ATTENTION

<table>
<thead>
<tr>
<th>NONE</th>
</tr>
</thead>
</table>

## KEY TEAM INTERDEPENDENCIES

- Gretchen task is Schedule (percentage 100%)  
- Task technical description (percentage 90%), and update (100%)  
- Domonique test plan (percentage 100%).
3. WEEKLY STATUS REPORT

To:  
Professor Rutherfoord

From:  
Domonique Jackson-Russell  
Gretchen Fahner

Subject: status
Period Ending: 03/01/2015
Self Assessment: Green, Yellow, Red

<table>
<thead>
<tr>
<th>ACTIVITIES COMPLETED THIS WEEK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed Deliverables:</td>
</tr>
<tr>
<td>➢ Requirement I</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACTIVITIES IN PROCESS</th>
<th>NEXT ACTION</th>
<th>DUE DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Software requirements (created in PolyRM), Paper prototype (draw by word)</td>
<td>➢ Finish Requirement II</td>
<td>➢ 03/01/2015</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACTIVITIES TO BE STARTED NEXT WEEK</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ System Design Documents</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LONG TERM PROJECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Requirement i-iii</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISSUES FOR IMMEDIATE ATTENTION</th>
</tr>
</thead>
</table>
| ➢ Gretchen please search for a program to create a PolyRm and paper prototype.  
➢ Update all documents on google drive. |

<table>
<thead>
<tr>
<th>KEY TEAM INTERDEPENDENCIES</th>
</tr>
</thead>
</table>
| ➢ Task software requirements (percentage 60%), and update (100%)  
➢ Domonique paper prototype (percentage 100%) update (100%). |
4. WEEKLY STATUS REPORT

To:
Professor Rutherfoord

From:
Domonique Jackson-Russell
Gretchen Fahner

Subject: status
Period Ending: 03/01/2015
Self Assessment: Green, Yellow, Red

ACTIVITIES COMPLETED THIS WEEK
Completed Deliverables:
- Requirement II

<table>
<thead>
<tr>
<th>ACTIVITIES IN PROCESS</th>
<th>NEXT ACTION</th>
<th>DUE DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Use Case Diagram, Use Case Flow, Class Diagram,</td>
<td>➢ Finish Requirement III</td>
<td>03/01/2015</td>
</tr>
</tbody>
</table>

ACTIVITIES TO BE STARTED NEXT WEEK
- System Design Documents

LONG TERM PROJECTS
- Technical Design

ISSUES FOR IMMEDIATE ATTENTION
- Update all your parts on google drive!!

KEY TEAM INTERDEPENDENCIES
- Gretchen task is Use Case Diagram and class Diagram (percentage 70%) update (100%),
- Task Case Flow of events (percentage 75%) update (100%)

5. WEEKLY STATUS REPORT

To:
Professor Rutherfoord

From:
Domonique Jackson-Russell
Gretchen Fahner

Subject: status
Period Ending: 03/01/2015
Self Assessment: Green, Yellow, Red

<table>
<thead>
<tr>
<th>ACTIVITIES COMPLETED THIS WEEK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed Deliverables:</td>
</tr>
<tr>
<td>➢ Requirement III</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACTIVITIES IN PROCESS</th>
<th>NEXT ACTION</th>
<th>DUE DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Class Diagrams (DIA), Class documentation (word), Entity relationship (drawing tool), decision tree.</td>
<td>➢ System Design Documentation</td>
<td>03/01/2015</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACTIVITIES TO BE STARTED NEXT WEEK</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ System Design Documentation.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LONG TERM PROJECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Technical Design</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISSUES FOR IMMEDIATE ATTENTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Update parts on google drive.</td>
</tr>
<tr>
<td>➢ Let team know if one needs any help it has to be due by 03/01!!</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KEY TEAM INTERDEPENDENCIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Gretchen task is Class Diagram (percentage 80%), update (100%)</td>
</tr>
<tr>
<td>➢ Task is class documentation (percentage 30%), update (100%) and</td>
</tr>
<tr>
<td>➢ Domonique task is Entity relationship and decision tree (percentage 60%) update (100%).</td>
</tr>
</tbody>
</table>
6. WEEKLY STATUS REPORT

To:
Professor Rutherfoord

From:
Domonique Jackson-Russell
Gretchen Fahner

Subject: status
Period Ending: 03/15/2015

Self Assessment: Green, Yellow, Red

<table>
<thead>
<tr>
<th>ACTIVITIES COMPLETED THIS WEEK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed Deliverables:</td>
</tr>
<tr>
<td>➢ Conceptual Design Documents</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACTIVITIES IN PROCESS</th>
<th>NEXT ACTION</th>
<th>DUE DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Report Format (word), Screen layouts/shots (prototype application)</td>
<td>➢ Technical Design</td>
<td>03/15/2015</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACTIVITIES TO BE STARTED NEXT WEEK</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Technical Design Documentation.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LONG TERM PROJECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Finish all system design documents.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISSUES FOR IMMEDIATE ATTENTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Update parts on google drive.</td>
</tr>
<tr>
<td>➢ Check over screen layout for any additional parts.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KEY TEAM INTERDEPENDENCIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Task is Report Formats (percentage 40%), update (100%)</td>
</tr>
<tr>
<td>➢ Domonique task is Screen layout (percentage 80%), update (100%)</td>
</tr>
</tbody>
</table>
7. WEEKLY STATUS REPORT

To:
Professor Rutherfoord

From:
Domonique Jackson-Russell
Gretchen Fahner

Subject: status
Period Ending: 03/29/2015
Self Assessment: Green, Yellow, Red

<table>
<thead>
<tr>
<th>ACTIVITIES COMPLETED THIS WEEK</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Completed Deliverables:</strong></td>
</tr>
<tr>
<td>- Technical Design</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACTIVITIES IN PROCESS</th>
<th>NEXT ACTION</th>
<th>DUE DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Class diagrams (all methods, attributes, and relationships in DIA), support text (word), database table.</td>
<td>- System Design Documentation</td>
<td>03/29/2015</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACTIVITIES TO BE STARTED NEXT WEEK</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Final Notebook</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LONG TERM PROJECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Project complete</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISSUES FOR IMMEDIATE ATTENTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Update parts on google drive.</td>
</tr>
<tr>
<td>- Go over any documents to include additional information.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KEY TEAM INTERDEPENDENCIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Gretchen task is detailed class diagrams and supporting text (percentage 60%), update (100%)</td>
</tr>
<tr>
<td>- Task is database table (percentage 30%), update (100%)</td>
</tr>
</tbody>
</table>
# 8. WEEKLY STATUS REPORT

To:
Professor Rutherfoord

From:
Domonique Jackson-Russell  
Gretchen Fahner

Subject: status
Period Ending: 4/12/2015
Self Assessment: Green, Yellow, Red

## ACTIVITIES COMPLETED THIS WEEK

<table>
<thead>
<tr>
<th>Completed Deliverables:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notebook</td>
</tr>
</tbody>
</table>

## ACTIVITIES IN PROCESS

<table>
<thead>
<tr>
<th>NEXT ACTION</th>
<th>DUE DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final project</td>
<td>4/12/2015</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACTIVITIES TO BE STARTED NEXT WEEK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final Project</td>
</tr>
</tbody>
</table>

## LONG TERM PROJECTS

| Project complete |

## ISSUES FOR IMMEDIATE ATTENTION

- Go over any documents to include additional information.
- Please update your minutes in google drive.

## KEY TEAM INTERDEPENDENCIES

- Domonique is task with Notebook (80%), update 100%
# Time Sheet

### Employee Information
- **Name:** Gretchen Fahner
- **Address:** [Department]
- **Phone:** [Phone]
- **Fax:** [Fax]
- **Email:** [Email]

### Pay Period
- **Starting:** 2/3/15
- **Ending:** 4/17/15

<table>
<thead>
<tr>
<th>PROJECT</th>
<th>Project Code</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
<th>Sat</th>
<th>Sun</th>
<th>Total Hrs</th>
<th>OT Hrs</th>
<th>Regular Hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>rt 1 (scope)</td>
<td></td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>8.00</td>
<td></td>
<td>8.00</td>
</tr>
<tr>
<td>rt 1 of 2 (schedule)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.00</td>
<td></td>
<td>3.00</td>
</tr>
<tr>
<td>Requirement I (QA)</td>
<td>FREE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.00</td>
<td></td>
<td>2.00</td>
</tr>
<tr>
<td>Requirement II (Use case&amp;Class diagram)</td>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.00</td>
<td></td>
<td>4.00</td>
</tr>
<tr>
<td>Requirement III (Class Diagram)</td>
<td></td>
<td>2</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.00</td>
<td></td>
<td>5.00</td>
</tr>
<tr>
<td>Conceptual Design (QA)</td>
<td>FREE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.00</td>
<td></td>
<td>4.00</td>
</tr>
<tr>
<td>Technical Design (detailed class and supporting text)</td>
<td></td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>8.00</td>
<td></td>
<td>8.00</td>
</tr>
<tr>
<td>Technical (QA)</td>
<td>FREE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.00</td>
<td></td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td></td>
<td>0.00</td>
</tr>
</tbody>
</table>

**Total Hrs:** 35.00 | **0.00** | **35.00**

**Notes:** Use increments of 0.25 (15 minutes) when recording hours.

---

**Employee Signature**

**Date**

---

**Supervisor Signature**

**Date**

**Supervisor Name:** [Name]
<table>
<thead>
<tr>
<th>PROJECT</th>
<th>Project Code</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
<th>Sat</th>
<th>Sun</th>
<th>Total Hrs</th>
<th>OT Hrs</th>
<th>Regular Hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>(data management)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>7.00</td>
<td></td>
<td>7.00</td>
</tr>
<tr>
<td>(def 2 (test plan))</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.00</td>
<td></td>
<td>4.00</td>
</tr>
<tr>
<td>(report (paper prototype))</td>
<td>FREE</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.00</td>
<td></td>
<td>4.00</td>
</tr>
<tr>
<td>(report II (QA))</td>
<td>FREE</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.00</td>
<td></td>
<td>3.00</td>
</tr>
<tr>
<td>(report III (Entity relationship &amp; decision tree))</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.00</td>
<td></td>
<td>5.00</td>
</tr>
<tr>
<td>(conceptual design (screen layout))</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7.00</td>
<td></td>
<td>7.00</td>
</tr>
<tr>
<td>(technical design (QA))</td>
<td>FREE</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.00</td>
<td></td>
<td>3.00</td>
</tr>
<tr>
<td>(book) (format doc and put in Appendix)</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.00</td>
<td></td>
<td>3.00</td>
</tr>
<tr>
<td>(overtime)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>(sick leave)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>(vacation)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>(personal leave)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>Total Hrs:</td>
<td></td>
<td>1.00</td>
<td>1.00</td>
<td>3.00</td>
<td>2.00</td>
<td>11.00</td>
<td>8.00</td>
<td>10.00</td>
<td>36.00</td>
<td>0.00</td>
<td>36.00</td>
</tr>
</tbody>
</table>

*Use increments of 0.25 (15 minutes) when recording hours.*

Employee: Domonique Jackson-Russell
Department: [Department]
Pay Period Starting: 2/3/15
Pay Period Ending: 4/17/15

April 2015

Calendar is for reference only.
# Timecard

**Employee**: Domonique Jackson-Russell  
**Department**: [Department]  
**Pay Period Starting**: 2/3/15  
**Ending**: 4/17/15

### April 2015

<table>
<thead>
<tr>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
<th>Sat</th>
<th>Sun</th>
<th>Total Hrs</th>
<th>OT Hrs</th>
<th>Regular Hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>7.00</td>
<td>7.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.00</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>FREE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.00</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>4.00</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.00</td>
<td>5.00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7.00</td>
<td>7.00</td>
<td></td>
</tr>
<tr>
<td>FREE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.00</td>
<td>3.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
</tbody>
</table>

- **Total Hrs**: 30.00  
- **OT Hrs**: 0.00  
- **Regular Hrs**: 30.00

**Notes**: Use increments of 0:25 (15 minutes) when recording hours.

---

**Employee Signature**  
**Date**

**Supervisor Signature**  
**Date**

**Supervisor Name**: [Name]
### Timecard

**Employee**: Mark Kordahl  
**Department**: [Department]  
**Pay Period Starting**: 2/3/15  
**Ending**: 4/17/15

<table>
<thead>
<tr>
<th>PROJECT</th>
<th>Project Code</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
<th>Sat</th>
<th>Sun</th>
<th>Total Hrs</th>
<th>OT Hrs</th>
<th>Regular Hrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part I (Team Org)</td>
<td></td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1.00</td>
<td></td>
<td>1.00</td>
</tr>
<tr>
<td>Part II (tech description)</td>
<td>1</td>
<td>1</td>
<td></td>
<td>3</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>6.00</td>
<td></td>
<td>6.00</td>
</tr>
<tr>
<td>Requirement II (software requirement)</td>
<td>3</td>
<td>1</td>
<td></td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>9.00</td>
<td></td>
<td>9.00</td>
</tr>
<tr>
<td>Requirement II (case flow)</td>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td>5.00</td>
<td></td>
<td>5.00</td>
</tr>
<tr>
<td>Requirement III (class doc)</td>
<td>2</td>
<td></td>
<td>1</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>5.00</td>
<td></td>
<td>5.00</td>
</tr>
<tr>
<td>Conceptual Design (Report Format)</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.00</td>
<td></td>
<td>1.00</td>
</tr>
<tr>
<td>Technical Design (database table)</td>
<td>FREE</td>
<td></td>
<td>1</td>
<td></td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td>5.00</td>
<td></td>
<td>5.00</td>
</tr>
<tr>
<td>Notebook(QA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>1.00</td>
<td></td>
<td>1.00</td>
</tr>
<tr>
<td>Holiday</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>Vacation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>Sick Leave</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>Personal Leave</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td></td>
<td>0.00</td>
</tr>
</tbody>
</table>

**Total Hrs**: 4.00  
**1.00**  
**1.00**  
**2.00**  
**6.00**  
**9.00**  
**10.00**  
**33.00**  
**0.00**  
**33.00**

*Note: Use increments of 0.25 (15 minutes) when recording hours.*

---

**Employee Signature**  
**Date**

**Supervisor Signature**  
**Date**

**Supervisor Name**: [Name]