SWE

Serve With Ease

Individual Contributions

Group 1 - Restaurant Automation

By: Christina Segerholm, Nishtha Sharma, Annie Antony, Athira Haridas, Emma Roussos, Christina Parry

Athira:

- Requirements Specification :
  - 100% for the following:
    - On Screen Requirements, System Architecture, System Design,, Hardware Requirement, Architectural Styles, Identifying Subsystems, Mapping Subsystems to Hardware, Persistent Data Storage, Network Protocol, Global Control Flow, Execution Orderliness, Time Dependency, Concurrency, User Effort Estimation - Changes we made for each of these
  - 50% of the following:
    - Summary of Changes, OCL Contracts, Data Structures
    - Use Case 1: Casual Description and Fully Dressed Description, System Operation Contracts, Interaction, Diagram, Design Patterns and Principles (under Interaction Diagrams), Changes we made
- Coding :
  - 50 % of the following:
    - HostTableScreen.java, LogInScreen.java,WaiterOneTicketScreen.java, WaiterTickListScreen.java

  100% of the following: IntegrationTestHost.java , ManagerScreen.java,
- Debugging :
  - 100% of the following:
    - HostTableScreen.java, LogInScreen.java,WaiterOneTicketScreen.java, WaiterTickListScreen.java, ManagerScreen.java
- Report Preparation :
  - Formatted Report 3 layout- 33%, delegated tasks amongst group-50%
- Other :
  - Overall revisions, readMeIntegrationTest - 80%

Annie:

- Requirements Specification :
  - 100% for the following:
    - Non-Functional Requirements, Data Types & Operation Signatures
    - Use Case 2: System Sequence Diagram, System Operation Contracts, Design Principles, Changes we made
  - 50% of the following:
    - Design of Test Cases, Summary of Changes
- Coding :
  - ChefOneTickScreen.java - 90% , ChefPanel.java - 50%, IntegrationTestWaiter.java-100%
- Debugging :
  - ChefInterface.java - 50%, ChefOneTickScreen.java - 90% , ChefPanel.java - 50%
- Report Preparation :
  - 100% of the following: Summary of Changes
- Other :
  - readMeIntegrationTest - 15%

Christina S:

- Requirements Specification :
  - 100% of the following:
    - Glossary of Terms, Mathematical Model, Domain Analysis: Domain Model Diagram, Concept Definitions, Association Definitions, Attribute Definitions, Algorithms - Changes we made for each of these
  - 50% of the following:
    - Use Case 1: Casual Description and Fully Dressed Description, System Operation Contracts, Interaction, Diagram, Design Patterns and Principles (under Interaction Diagrams), Changes we made for each of these

        - OCL Contracts
    - 90% of the following: Future Work
- Software Design :
    - 100% of the following:
        - Designed overall layout of system and architecture, creator of distributed databases and message controller, Major Ideas behind System Design and Database distribution, Created GUI template that all interfaces would use.
- Coding :
    - 100% of the following packages:
        - Configuration, DatabaseA, DatabaseB, DatabaseC, MessageController,
    - 100% of the following classes:
        - ChefInterface.java, ChefMessageListener.java, ChefMessageSEnder.java, HostInterface.java, HostMessageListener.java, HostMessageSender.java, TabletApp.java, ManagerInterface.java, ManagerMessageListener.java, ManagerMessageSender.java, DatabaseLoadFRomFileTesting.java,MessageTesting.java, TicketToMessageConversion.java, DatabaseCListener.java, DatabaseCSender.java, DishForTicket.java, KeypadScreen.java, WaiterInterface.java, WaiterMessageListener.java, WaiterMessageSender.java
    - 50 % of the following classes:
        - HostTableScreen.java, LogInScreen.java,WaiterOneTicketScreen.java, WaiterTickListScreen.java
- Debugging :
    - 100% of the following:
        - All backend work (message controller, databases, message sending, ticket sending, interface communication)
- Report Preparation :
    - Formatted Report 3 layout- 33%, delegated tasks amongst group-50%
- Other :
    - 100% of the following:
        - readme.txt, Technical Document, readmeUnit.txt, readMeData.txt, Organized and Maintained github repository
    - Overall Revisions, readMeIntegrationTest-5%, skits-50%

Emma:

- Requirements Specification :
    - 100% for the following:
        - Use Case 3 : Casual Description, System Operation Contracts, Interaction Diagram, Design Patterns and Principles (under Interaction Diagrams), System Sequence Diagram
    - 50% for the following:
        - Design of Test Cases, Use Case 3: Fully Dressed Description, Changes we made
- Report Preparation:
    - 100% for the following:
        - Inserted all information having to do with Use Case 3

Christina P:

- Requirements Specification :
    - 100% of the following:
        - Traceability Matrices and Descriptions, Project Management and History of Work, Demo 2 Brochure, Revised CSR
    - 50% of the following:
        - Use Case 3 : Changes to Use Case 3, Fully Dressed Description, Design of Test Cases
    - Added to Summary of Changes
- Report Preparation :

    - Revisions on CSR and Use Case 3, Formatting

Nishtha:

- Requirements Specification :
    - 100% of the following :
        - Design Patterns , Key Accomplishments , Functional Requirements , Class Diagrams and Interface Specification (Class Diagram, Key, Descriptions) , User Interface Design and Implementation, Use Case 2: Casual Description and Fully Dressed Description, Interaction Diagram, Design Patterns (under Interaction Diagrams)
    - Data Structures - 50%
    - Future Work - 10%
- Coding :
    - ChefPanel.java - 50%, ChefOneTickScreen.java - 10%
- Debugging :
    - ChefInterface.java - 50% , ChefOneTickScreen.java - 10% , ChefPanel.java - 50%
- Report Preparation :
    - 100% of the following :
        - Labeled and captioned all Figures in report , Presentation for Demo 2 , UML Diagrams document , User documentation , Formatted OCL Contracts into tables, Formatted On-Screen Requirements
    - 50 % of the following :
        - ReadMe.txt for Demo 2 , Skits
    - Formatted Report 3 overall layout - 33%
- Other : Overall Revisions

Note we did not have any Data Collection Scripts. This is because we have not implemented the SQL part of the project yet. We did still do the documentation on it. Right now, the three databases are implemented as java classes (DatabaseAController, DatabaseBController, DatabaseCController).