Implementation Document

for

Shop Inventory Manager

Version 1.2

Prepared by

Group #: 20

Kshitij Kabeer	180366	kshitijkabeer@gmail.com
Rishabh Kothary	180608	rishabhkothary76@gmail.com
Kartavya	180343	kartavya4301@gmail.com
Pravar Deep Singh	160508	pravardeepsingh@gmail.com

Course:	CS253
Mentor TA:	Swastik Maiti
Date:	28th April 2022

CONTENTS II

REVISIONS II

1 INTRODUCTION 1

2 UNIT TESTING 2

3 INTEGRATION TESTING 3

4 SYSTEM TESTING 4

5 CONCLUTION 5

APPENDIX A - GROUP LOG 6

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.1	Kshitij Kabeer Rishabh Kothary	Initial Draft	21/03/2022
1.2	Kshitij Kabeer	Minor Changes	28/04/2022

1 Implementation Details

Provide the details of programming languages, frameworks, libraries, database systems, build systems, etc. that you have used for implementing your software.
Provide a brief justification of choosing any tool by stating its benefits over the alternatives.

Programming Language: C++

- We chose C++ because it is a highly versatile language and very fast. Also, our team mostly had expertise in C++.

Frameworks: Qt for Front-end

- Qt Creator has GUI tools for quickly making a front end interface
- The reason we used Qt over visual studio is that it can be integrated seamlessly with other build systems like CMake, and also Qt is cross-platform unlike visual studio.
-

Libraries: Google OR-Tools (for optimization), SQLite3 (for accessing Database)

- Sqlite3 is a simple library that helps maintain offline databases, and supports passing SQL commands via strings
- Google OR-Tools provides a large range of optimization tools and frameworks and is easy to use.

Build system: CMake, with Mingw-x64 Make

- CMake is a very powerful build system that supports cross platform compilation of code. It takes care of all dependencies and circumvents the need to write complicated Makefiles.
- Mingw-x64 build tool is easy to use and is similar to gcc and g++ on Linux, which means that there will be minimal changes required when porting this code over platforms

2 Codebase

Provide the link to your github repository.
Mention briefly how to navigate the codebase.

https://github.com/Kshitij1K/ShopInventoryManager

The Codebase is divided into multiple CMake Packages. The main executable is in the Main package. Each CMake package has to be built individually, I am still working on automating the process. Each package has an include folder that has the class and function declarations and a src folder that has the definitions for all these class functions.
To use or-tools x64 Naive Command Prompt of Visual Studio is required and make function is needed to use Makefile in the or-tools directory. Install the library externally as the size of library is large. To run code related to library go to the or-tools directory and use
"make run SOURCE=relative_path_to_the _file_to_be_run" and to compile use
"make build SOURCE=relative_path_to_the _file_to_be_run" where the executable is saved in the bin directory of the or-tools library. Go through the MAKEFILE of or-tools for deeper understanding.

For connecting Sqlite3 database to c++ I installed sqlite3 package . The routines and functions available in the package are used to connect to Database ```Db1test``` which further allows storing and extracting data .
To use the above package compile the database along with linker file sqlite3 ```g++ database.cpp -l sqlite3 ```.
This code is compatible for windows OS.

Installation Instructions have been written just before the beta testing phase, so you can refer to those to get updated instructions for installation.

3 Completeness

Provide the details of the part of the SRS that have been completed in the implementation.

Provide the future development plan by listing down the features that will be added in the (may be hypothetical) future versions.

We have a first version of Optimization tool
Basic packages framework
Successfully integrated Qt
Successfully done the part of DBMS and wrote all the functions to update the DBMS through C++.
Linked the DBMS to C++ using SQLite version 3.

Appendix A - Group Log

<*Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to implement your software*>

Pravar and Kartavya : Working on Database functions

Rishabh - Working on Optimization

Kshitij - Working on State Machine, Qt, EventHandler, Database boilerplate code, Database functions, Integration.