**NYU Tandon School of Engineering**

Computer Science and Engineering

**Smart Café**

Software Engineering Standards

Requirements/Analysis Specification

**Version 1.0**

Document Number: RAS-001

**Team A1**

**Samuel Huang (sh3244)**

**Ryan Goudjil (rg2398)**

**John J. Simon (jjs597)**

**Chia-Ching Song (ccs401)**

Review and Approvals

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Function** | **Date** | **Signature** |
| Samuel Huang | Author |  | SH |
| Ryan Goudjil | Author |  | RG |
| John J. Simon | Author |  | JJS |
| Chia-Ching Song | Author |  | CS |
|  | Reviewer |  |  |
|  | Reviewer |  |  |

Revision Level

|  |  |  |
| --- | --- | --- |
| **Date** | **Revision Number** | **Purpose** |
|  | Version 1.0 | Initial Release |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table of Contents

[1. Introduction 6](#_Toc443905407)

[1.1. Purpose 6](#_Toc443905408)

[2. Scope 6](#_Toc443905409)

[2.1. Identification 6](#_Toc443905410)

[2.2. Bounds 6](#_Toc443905411)

[2.3. Objectives 6](#_Toc443905412)

[2.4. System Overview 7](#_Toc443905413)

[2.5. Document Overview 7](#_Toc443905414)

[3. Reference Documents 7](#_Toc443905415)

[4. Business Requirements 7](#_Toc443905416)

[4.1. Technology 7](#_Toc443905417)

[4.2. Economics 7](#_Toc443905418)

[4.3. Regulatory and Legal 8](#_Toc443905419)

[4.4. Market Considerations 8](#_Toc443905420)

[4.5. Risks and Alternatives 8](#_Toc443905421)

[4.6. Human Resources and Training 8](#_Toc443905422)

[5. Descriptive Functional and Non-Functional Requirements 8](#_Toc443905423)

[5.1. Functional Descriptive Detailed Requirements 8](#_Toc443905424)

[5.2. Requirement Uses Case 8](#_Toc443905425)

[5.3. Non-Functional Descriptive Detailed Requirements 9](#_Toc443905426)

[5.4. Context Diagram 9](#_Toc443905427)

[6. Functional Requirements Analysis Specifications 9](#_Toc443905428)

[6.1. Systems Capabilities Requirement 9](#_Toc443905429)

[6.1.1. Capabilities 10](#_Toc443905430)

[6.2. User Interface Requirements 10](#_Toc443905431)

[6.3. Component Architecture 10](#_Toc443905432)

[6.3.1. Component Descriptions 10](#_Toc443905433)

[6.3.2. Component Architecture Diagrams 11](#_Toc443905434)

[6.4. Class Diagrams 11](#_Toc443905435)

[6.5. Class Relationship/Interactions 12](#_Toc443905436)

[6.6. Event Section 13](#_Toc443905437)

[6.6.1. Event Dictionary 13](#_Toc443905438)

[6.6.2. Events Diagram 13](#_Toc443905439)

[6.7. Activity/State (Scenario) Section (To be completed in Design) 13](#_Toc443905440)

[6.7.1. Activity (Scenario) Diagrams 13](#_Toc443905441)

[6.7.2. Activity (Scenario) Specifications 13](#_Toc443905442)

[6.8. Sequence Diagrams 13](#_Toc443905443)

[6.9. Collaboration Diagrams 14](#_Toc443905444)

[6.10. Dictionaries 14](#_Toc443905445)

[7. Non-Functional / Operational Specifications 14](#_Toc443905446)

[7.1. System External Interface Requirements 14](#_Toc443905447)

[7.2. Safety Requirements 14](#_Toc443905448)

[7.3. Security and Privacy Requirements 14](#_Toc443905449)

[7.4. System Environment Requirements 14](#_Toc443905450)

[7.5. Computer Resource Requirements 14](#_Toc443905451)

[7.6. System Quality Factors 15](#_Toc443905452)

[7.7. Design and Construction Constraints 15](#_Toc443905453)

[7.8. Personnel-Related Requirements 15](#_Toc443905454)

[7.9. Training-Related Requirements 15](#_Toc443905455)

[7.10. Logistics-Related Requirements 15](#_Toc443905456)

[7.11. Packaging Requirements 15](#_Toc443905457)

[7.12. Precedence and Criticality Requirements 15](#_Toc443905458)

[*7.13.* Other Requirements 15](#_Toc443905459)

[8. System Test Plan Requirements 15](#_Toc443905460)

[9. Qualification Provisions 16](#_Toc443905461)

[10. Requirements Traceability 16](#_Toc443905462)

[11. Rationale 16](#_Toc443905463)

[12. Notes 17](#_Toc443905464)

[13. Appendices 17](#_Toc443905465)

[13.1. Dictionary 17](#_Toc443905466)

[13.2. UML Diagrams 17](#_Toc443905467)

[13.3. Schedule Tracking 17](#_Toc443905468)

[13.4. Defect Tracking 17](#_Toc443905469)

# Introduction

## Purpose

The purpose of this RAS is to delineate the requirements for our smart/connected coffee machine project and describe the means of accomplishing necessary product function in an organized document. This project’s requirements, including the functional Android UI/Application, the communications and command server/database, and the embedded system in the coffee machine will be plan-implemented and detailed in this document. Immediate business and technology factors, risks, and solutions will be analyzed. This document is intended to verify that the project possesses a plan of development and have met all the theoretical requirements for completion in accordance to Smart Café’s business.

# Scope

## Identification

Smart Café Software Analysis Specification (SAS), Version 1.0, November 30, 2015

## Bounds

The system is bounded by the application, the server, and the coffee machine controller with interaction between the application and server, and the server and coffee machine controller, and the coffee machine controller with the coffee machine. The chain of interaction takes the user’s input through the application and deciphers it into a collection of commands and features.

## Objectives

The Smart Café is being designed in to order to provide users with a more convenient and less expensive way of obtaining coffee.  Currently people have to stop and make the time to run and buy coffee or prepare and wait on a pot to brew.  With the Smart café, the users (coffee drinkers) will gain a seamless connection to their own personal coffee maker.  Smart Café integrates an easy-to-use phone application to allow the user instant and continuous access to their coffee machine whenever and wherever they may be.   
  
In order to meet the needs of the client, the Smart Café requires a server to connect and maintain service between the user’s phone application and the microcontroller within the coffeemaker.  The users will be provided with a simple and clean application user interface allowing them to start the coffee process with the press of a button.  To make the app more personal for each user, the application will keep track of your daily and average coffee intake as well as offering the ability to set a timed schedule for coffee throughout the day.  Equipped with notifications, users will know exactly when their coffee is ready and will be asked to confirm a scheduled coffee to make sure it is still wanted.

**Project Deliverables**

#### Project proposal Software Requirement Specification (SRS)

##### 10/13

#### Software Project Management Plan (SPMP)

##### 11/4

#### Software Analysis Specification (SAS)

##### 11/30

#### Software Design Document (SDD)

##### 12/2

## System Overview

The system includes the application, the controller, the coffee machine, and the user.

## Document Overview

The document includes the application, the controller, the coffee machine, and the user.

# Reference Documents

Song, Chia-Ching, Samuel Huang, John Simon, and Ryan Goudjil. Project Proposal. NYU Tandon School of Engineering: Team A1, 30 Nov. 2015. Docx.

Song, Chia-Ching, Samuel Huang, John Simon, and Ryan Goudjil. Software Project Management Proposal. NYU Tandon School of Engineering: Team A1, 30 Nov. 2015. Docx.

Song, Chia-Ching, Samuel Huang, John Simon, and Ryan Goudjil. Software Analysis Specification. NYU Tandon School of Engineering: Team A1, 30 Nov. 2015. Docx.

# Business Requirements

## Technology

The technology driving this project is the ability to connect multiple devices through a central common use case, the smart phone. This system of connectivity allows for the existence of the “smart home” or the “connected home”.

## Economics

The market drivers are that there are high end home products that allow for control and connectivity through a user’s smart phone. They exist, but are very expensive as well as don’t work well according to customer reviews.

## Regulatory and Legal

N/A

## Market Considerations

The business market that’s driving this system is the emerging

The considerations we need to make in terms of the market is the fact that there are coffee machines that come with apps the user can download already on the market. However, these are only for high end coffee machines and customer reviews show dissatisfaction with these apps.

## Risks and Alternatives

Risks for this project is that only one member of the team has a greater understanding of databases. The solution is that two more members are learning databases as they are working on the project. Another risk of the system is interfacing with the hardware. Meetings with the hardware team to figure out how the code for the interfacing is documented will be the solution for this risk.

## Human Resources and Training

The resources and training needed for this project is in android development, as well as training for databases for a few of the members. This means training in android studio using java, as well as training in building server applications in C++.

# Descriptive Functional and Non-Functional Requirements

## Functional Descriptive Detailed Requirements

#### Activates the coffee machine.

#### Activation can be done when the user inputs the command, or schedules the activation.

#### Deactivates the coffee machine.

#### Keeps track of user’s caffeine and calorie intake of coffee.

## Requirement Uses Case

#### User Application (Front End)

##### Set schedule for making coffee at certain times

##### Specify the amount of coffee to prepare

##### Send confirmation to make coffee now or in “X” minutes

##### Ability to cancel process

##### Choose whether to prepare coffee or tea

#### Back End

##### Connect the application to central processing server

##### Communicate with server

##### Keep track of daily coffee intake

#### Server

##### Receive commands from user

##### Send command to user corresponding coffee machine

##### Use database to store user profiles

##### Query database for stored info

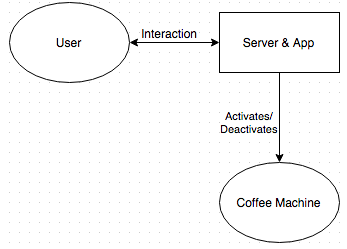
#### Database

##### Store user profile logically

## Non-Functional Descriptive Detailed Requirements

#### Not applicable

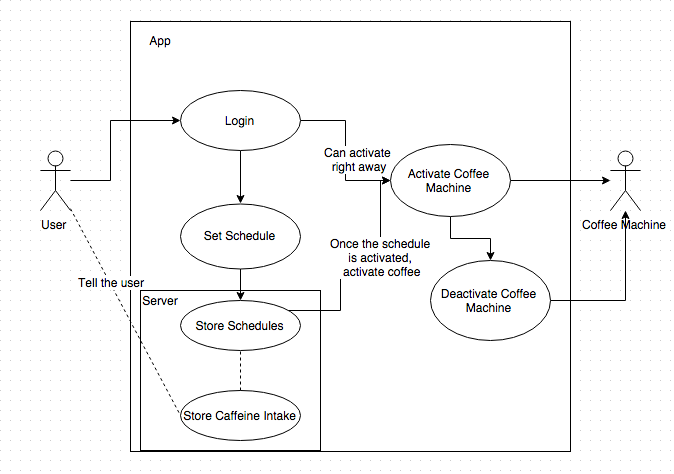
## Context Diagram



# Functional Requirements Analysis Specifications

## Systems Capabilities Requirement

### Capabilities



## User Interface Requirements

Android Studio will be used to create the user interface of the application. The user interface will require:

#### Main log in for users: username and password required or create a new account

#### Creating a new account screen taking username, password, and machine ID

#### Screen for making a coffee order immediately or in “x” minutes

#### Screen for scheduling coffee at set times and days

#### Screen for checking user info as well editing

## Component Architecture

### Component Descriptions

There are three internal subsystems within the system:

#### Phone application:

##### This is the interface for the user to interact with the system

##### Allows the user to create an account with their coffee machine and update user information

##### Allows the user to set/update coffee alarms

#### Allows user to initiate coffee making

#### Server:

##### Allows communication between application, database, and the coffee machines

##### Receive user requests for coffee and activates respective coffee machine

##### Sends user account information including coffee intake, alarms, and any updated information to database

##### Retrieves alarm data from database to activate coffee machine and sends user notifications

##### Receives incoming data from application to execute commands

#### Database:

##### Receives data from server to store about users

##### Stores user information including alarms, coffee intake, and updates

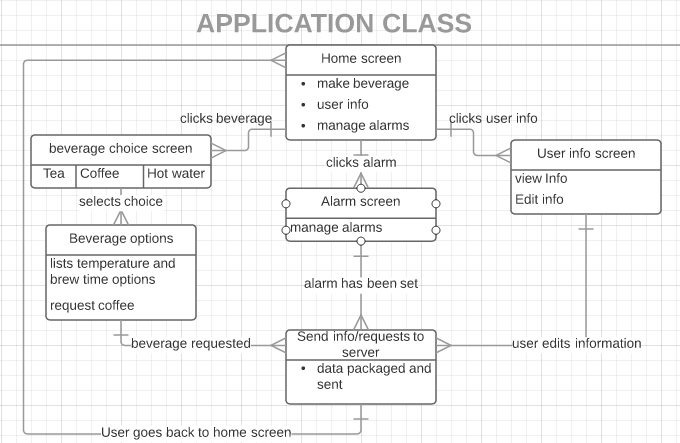
### Component Architecture Diagrams

#### The phone application serves as the interface between the user and the system

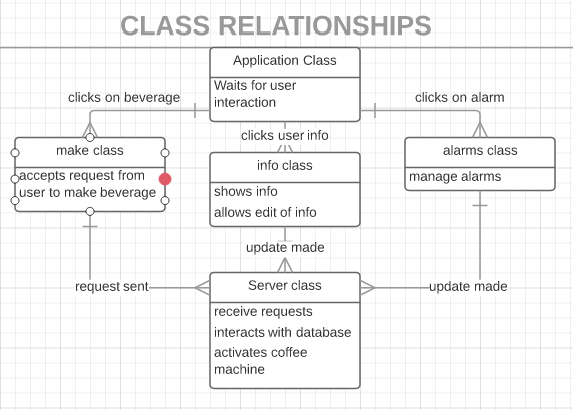
#### The server acts as the communication link between the application and the coffee machine

#### The database stores user information and is accessed only by the server

## Class Diagrams



## Class Relationship/Interactions



## Event Section

#### User makes request for coffee

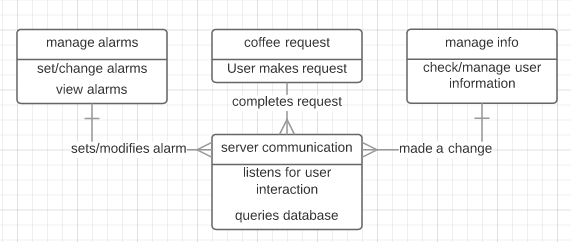
#### User checks/manages user information and alarms

#### Server queries database for current alarms

#### Server receives request for coffee and activates machine

### Event Dictionary

### Events Diagram



## Activity/State (Scenario) Section (To be completed in Design)

Will be included once completed

### Activity (Scenario) Diagrams

### Activity (Scenario) Specifications

## Sequence Diagrams

In Appendix.

## Collaboration Diagrams

Will be included after the Design Life Cycle.

## Dictionaries

In Appendix (see 13.1).

# Non-Functional / Operational Specifications

## System External Interface Requirements

#### Server must be able to connect to coffee machine

#### Users must be able to download app on their phones

## Safety Requirements

Not Applicable

## Security and Privacy Requirements

Not Applicable

## System Environment Requirements

System must be able to connect to the internet for network communications

## Computer Resource Requirements

Some additional resources may be required later on:

#### **Computer Hardware Requirements**

#### RaspberryPi’s needed to run the server

#### **Computer Hardware Resource Requirements**

#### Not Applicable

#### **Computer Software Requirements**

#### AndroidStudio and Xcode required

#### **Computer Communications Requirements**

Server must be able to communicate with user phone applications and coffee machines

## System Quality Factors

#### Network must be accessible 95% of the time

#### Coffee requests must go through 100% of time

## Design and Construction Constraints

Currently the software must be able to communicate with the smart coffee maker that is also in development, with the possibility of being able to communicate with smart coffee makers from other manufacturers in the future.

## Personnel-Related Requirements

Not applicable

## Training-Related Requirements

Not applicable

## Logistics-Related Requirements

Not applicable

## Packaging Requirements

Not applicable

## Precedence and Criticality Requirements

Not applicable

## Other Requirements

Not applicable

# System Test Plan Requirements

While developing modules, said modules will be tested by someone other than the module’s programmer. The app will be simulated in whichever phone simulator the chosen IDE comes with. The app will also be tested on an actual phone once development advances enough. The app will be tested by members assigned with testing responsibilities. Once the module has been tested, the member that tested the module will submit proper documentation. One tool that will be used is Appium, an open source testing framework that allows the members assigned to testing to script automatic app testing. Appium will be used later when the app is closer to being finalized. For the database, WAMP will be used to test it. As said in the SRS all testing will be documented, and any revisions and documentation history will be available to all members.

# Qualification Provisions

The sections were divided evenly between the team members and deadlines for progress were held (2-day section cycles). Before commencing the required work, the team collectively analyzed the document requirements to ensure clarity on the necessities and expectations. To maintain quality, the team had peer reviews after sections were completed, and individual members revised the sections based on the results of the review. Afterwards, the team members held an inspection meeting to ensure that the revisions have resolved all issues and that no new issues had appeared.

The document will be reviewed by all members of the team, as well as be presented to at least two outside persons not in the development team to check for defects as well as comprehensibility in order to verify correctness. The document will keep a record of the changes made within itself to maintain traceability. Modification will be done through the editable form of the document, and versions will be presented in a finalized version upon completion. Before the presentation of the document, all members must be reviewed once more by all members of the team, and one person on the team will conceive and direct the final formatting and presentation style to ensure completeness and consistency.

With regards to the purpose of the RAS specifically, the team has collectively been brought up to date with the plan of development through definitively selected development environments and hardware. Each functional module’s hardware, development environment, and language have been determined and agreed upon, and the presented plan has been researched and understood at an in-depth level in order to ensure capability and coherence proceeding with the plan.

# Requirements Traceability

Phone application/user interface is necessary for the user to communicate with the coffee maker. In order for the communication to take place, a central server is required to handle and process all logic and integration between application and the machine. A database is required to contain user info to discern respective machines with users and communicate data back to user application.

Finally, an interface is necessary within the coffee machine to make accept commands from the central server (which come from the phone application) and then send notifications of progress back to the server and then back to the user.

# Rationale

Not applicable at this time.

# Notes

N/A

# Appendices

## Dictionary

Fdas

## UML Diagrams

Identified in body of document.

## Schedule Tracking

Fdsa

## Defect Tracking

Fdas