Automatic Time Management System
(ATOMS)

CS 3337 Software Engineering

Application Requirements Specification Document

Prepared By:

LaFrance, Montague
Chan, Micky
Gomez, Carlos
Vargason, Austin

Date:
April 20, 2019

Table of Contents

Document Revision History

Version #	*Revision Author*	*Revision Summary*	*Revision Date*
0.1	Austin Vargason	Initial Draft	2/23/19
0.2	Austin Vargason	Meeting Template Requirements	3/9/19
0.3	Carlos Gomez	Added Performance and Safety Requirements	3/13/19
0.4	Austin Vargason	Completed Section 2 and 3	3/14/19
0.5 - 0.7	Austin Vargason	Completed Initial Document Submission	3/15/19
0.8	Austin Vargason	Cleanup of Requirements	3/24/19
0.9	Carlos Gomez, Micky Chan, Austin Vargason, Montague LaFrance	Document Review and DFD addition	3/30/19
1.0	Refactoring Document	Austin Vargason	4/20/19

1.0 Introduction

This document seeks to outline the software requirements for the Automatic Time Management System (ATOMS) android application. ATOMS is an Android Application that helps the user automatically manage their time.

1.1 Purpose

The purpose of this SRS document is to specify the requirements for the ATOMS application.

1.2 Intended Audience and Reading Suggestion

This document is intended to be viewed by developers involved with the ATOMS application to fully understand the requirements of ATOMS.

1.3 Product Scope

The scope of the ATOMS application is limited to necessary use required by this class.

1.4 Definitions and Acronyms

- Android Studio: Development environment to develop Android Applications
- Java: Object Oriented Programming Language
- GUI: Graphical User Interface
- ATOMS: Automatic Time Management System
- API: Application Programming Interface
- OS: Operating System
- XML: extensible markup language
- Strictly Scheduled Event: A calendar event that has a pre-defined start and end time
- Automatically Scheduled Event: A calendar event scheduled by the defined ATOMS algorithm

1.5 References

1.5.1 Google Sign-In and Google Play Services Authentication
Source: https://developers.google.com/identity/sign-in/android/start-integrating

1.5.2 Android Calendar Provider
Source: https://developer.android.com/guide/topics/providers/calendar-provider

1.5.3 Google OAuth Client
Source: https://developers.google.com/api-client-library/java/google-oauth-java-client/

1.5.4 Firebase Application Integration
Source: https://firebase.google.com/docs/android/setup

1.5.4 Google Code Labs for Android Development
Source: https://codelabs.developers.google.com/android-training/

1.5.5 Android Studio Documentation
Source: https://developer.android.com/studio/intro

2.0 Overall Description

The ATOMS application is an Android application for use on Android phones at or above OS level 7.1 to be used for time management and automatic scheduling.

2.1 Product Perspective

ATOMS is dependent on the larger structure of Firebase integration and Google Sign In in order to maintain user states and usage statistics.

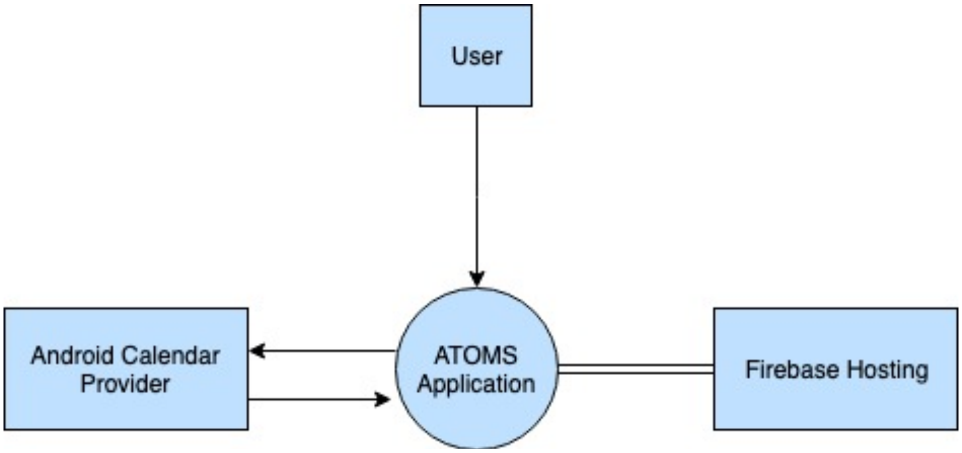
Atoms differs from available software on the market through its automatic scheduling capabilities and general ease of use.

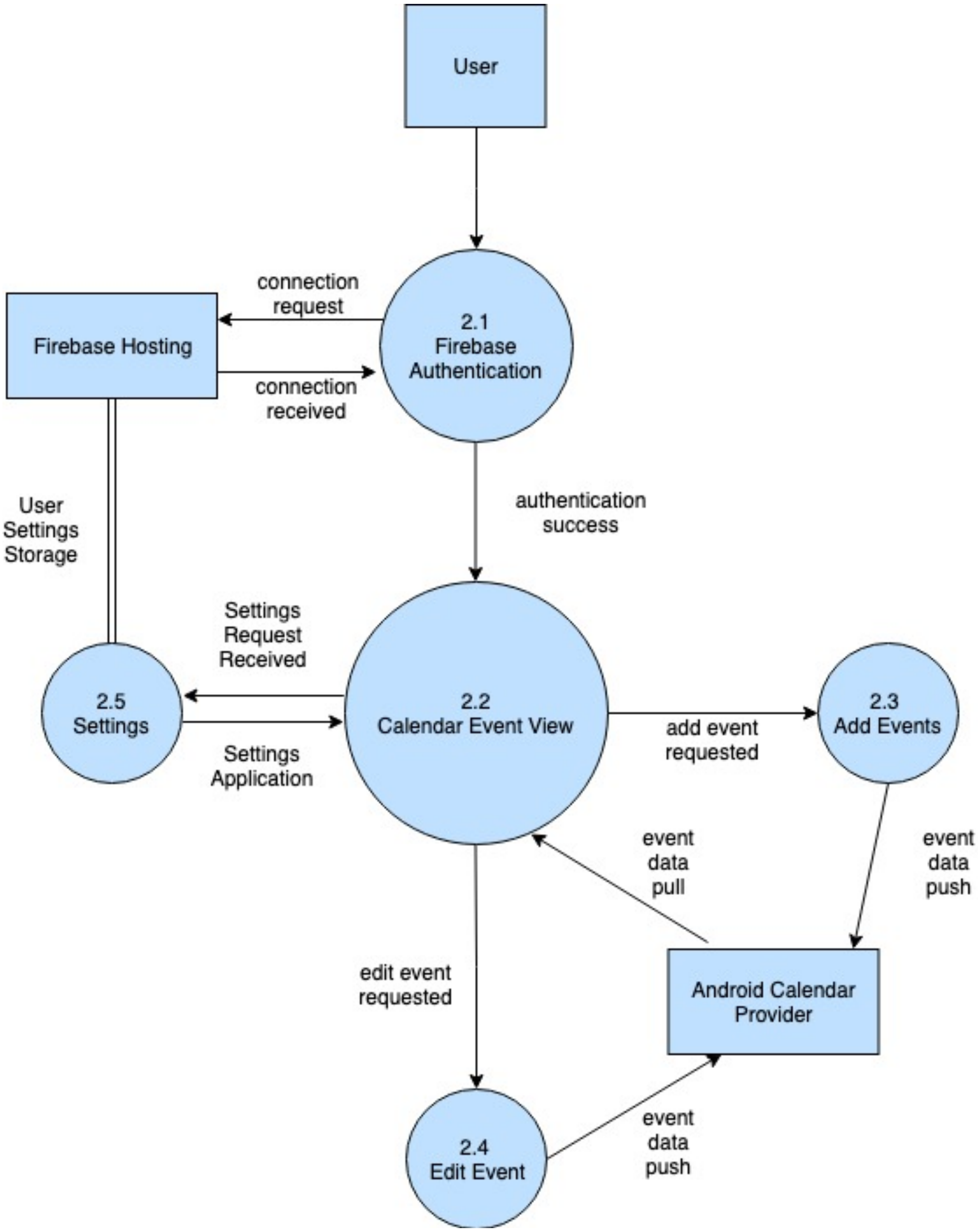
2.2 Product Functions

The major features of ATOMS are summarized below:

- **Login**: ATOMS users may log in through their existing Google Account.
- **Calendar Views**: ATOMS can present calendar events in a daily, weekly, or monthly calendar view.
- **Automatic Event Scheduling**: ATOMS can automatically schedule calendar events, based on the user's current schedule.
- **Theme Options**: ATOMS is customizable to a user's color preference.
- **Gesture View Switching**: Swiping left or right on the ATOMS main screen will change the date range for the Calendar View.
- **Data Backup**: All calendar changes shall be applied to the user's underlying calendar data.

2.2.1 High-Level DFDs





2.3 User Classes and Characteristics

Not Applicable.

2.4 Operating Environment

The ATOMS application requires Android version 7.1 and above.

2.5 Design and Implementation Restraints

Not Applicable.

2.6 User Documentation

Not Applicable.

2.7 Assumptions and Dependencies

- API Dependencies: Google Calendar, Google Sign In, and Firebase Integration Dependencies
- This application uses a minimum API version of Android presented in 2.4 Operating Environment

2.8 Apportioning of Requirements

Requirements are apportioned into a per module (Section 2.2.1) basis defined in section 4.0.

3.0 External Interface Requirements

3.1 User Interfaces

User Interface described in detail in section 4.0 per module.

3.2 Hardware Interfaces

Not applicable.

3.3 Software Interfaces

3.3.1 Google Sign-In and Google Play Services Authentication
Source: https://developers.google.com/identity/sign-in/android/start-integrating

3.3.2 Android Calendar Provider
Source: https://developer.android.com/guide/topics/providers/calendar-provider

3.3.3 Google OAuth Client
Source: https://developers.google.com/api-client-library/java/google-oauth-java-client/

3.3.4 Firebase Application Integration
Source: https://firebase.google.com/docs/android/setup

3.4 Communication Interfaces

- Google Sign In is handled through Google Play Services API and Google Cloud Platform Web Client.
- Firebase and Oath communicates via REST API communication methods.

4.0 Requirements Specification Per Module

Those requirements represented using the language “shall” will be implemented into the final application. Those using the language “may” should only be implemented given that time restraints permit. The Requirements in this section shall be split into the following modules: Firebase Authentication, Calendar Event View, Add Events, Edit Events, and Settings.

Module 2.1: Firebase Authentication

Requirement Number	*Requirement Description*
2.1.1	The login screen shall be presented to the user when previous authentication is not present.
2.1.2	The login screen shall present the ATOMS logo at the top of the layout
2.1.3	The login screen shall present a button to login with Google Sign In implemented in the Google Sign In API.
2.1.4	The login screen shall only redirect to the main app upon proper retrieval of a Google Authentication and Firebase Connection.
2.1.5	The Login Screen shall exclusively use Google Sign-In for a Firebase Authentication.

Module 2.2: Calendar Event View

Requirement Number	*Requirement Description*
2.2.1	The center of the main app screen shall display each calendar event in a card layout.
2.2.3	The default view for events on the main screen shall be the daily view.
2.2.4	The main app screen shall display a floating action button in the right corner of the screen to add a new event to the current view.
2.2.5	The main app screen shall have a settings menu in the upper right-hand corner of the screen presenting the items: "Settings", and "Log Out".
2.2.6	The main app screen shall be easily navigable and scroll when necessary.
2.2.7	Calendar events shall be loaded from the Android Calendar Provider upon load of the main app screen and selection of a new view type.
2.2.8	The main app screen shall consist of sidebar that presenting the view choice items: "Daily", "Weekly", and "Monthly".
2.2.9	Event views shall update the main screen to a new view type (weekly, monthly, daily).
2.2.10	The user shall be notified that changes in the app to calendar events will affect their underlying Google Calendar through the Android Calendar Provider.
2.2.11	The Floating Action Button on the main app screen shall redirect to the Add Events module.
2.2.12	The Settings menu item "Settings" shall redirect to the Software Settings Module.
2.2.13	A tap of an Event on the main module screen shall redirect to the Edit Event Module.

Module 2.3: Add Events Module

Requirement Number	*Requirement Description*
2.3.1	The Add Events module shall be launched upon selection of the floating action button in the Main Module.
2.3.2	The Add Events module's UI shall present a form to the User to gather information about an event that will be added to the Main Module.
2.3.3	The form specified in requirement: **2.3.2** shall contain a dropdown box to select a Calendar Event type ("auto-scheduled" or "standard").
2.3.4	The form specified in requirement: **2.3.2** shall be dynamically updated based on the dropdown box item selection.
2.3.5	The form generated for "auto-scheduled" events shall consist of form options to gather: When the event should start being scheduled, when the event should be completed by, and the estimated number of hours to complete the event.
2.3.6	The form generated for "standard" events shall consist of form options to gather: the event start time, the event end time, whether it repeats (if the event repeats ATOMS shall request the days for which it repeats and when the repetition ends).
2.3.7	"auto-scheduled" events shall be handled by the Add Events module and a schedule shall be generated then outputted to the Android Calendar Provider.
2.3.8	"standard" events shall be handled by the Add Events module and outputted to the Android Calendar Provider.

Module 2.4: Edit Events Module

Requirement Number	*Requirement Description*
2.4.1	The Edit Events module shall be launched upon a press of a calendar event in the main module
2.4.2	The Edit Events Module shall take the long-held event as input into a form where event properties shall be edited.
2.4.3	The form specified in requirement: **2.4.2** shall contain all the properties of the event (start time, end time, title, priority, and repetitions) that shall be edited.
2.4.4	Upon submission of the updated event data, the event shall be updated and outputted to the main module.

Module 2.5: Settings Module

Requirement Number	*Requirement Description*
2.5.1	The settings module shall be launched upon selection of the settings menu item: “Settings”.
2.5.2	The Settings Module shall present settings for the app including: App Theme, Time Format, and Date Format.
2.5.3	User settings shall be saved and outputted to firebase save data.
2.5.4	Theme selections shall consist of predefined app themes and update the general color theme of the app accordingly.
2.5.5	Time format options shall update the displayed time format of events in the main module.
2.5.6	Date Format options shall update the date format (start of the week, general date display format) in the main module.
2.5.7	The settings module shall present predefined theme choice options of “Dark Mode”, “Light Mode”, “Classic Android”, or “Oceanic”

4.1 External Interfaces Requirements

Requirement Number	*Requirement Description*
4.1.1	The ATOMS application shall interface with Firebase for storing user Google Accounts and settings.
4.1.2	The ATOMS application shall interface with Google Sign In.
4.1.3	The ATOMS application shall interface with the Android Calendar Provider to deal with user’s calendar data.

4.2 Logical Database Requirements

Requirement Number	*Requirement Description*
4.2.1	Firebase online database shall store user’s settings preferences based on a user id.
4.2.2	Firebase online user storage shall store all previously authenticated user’s to ATOMS and when they last logged in.

4.3 Design Constraints

- **OS Level Constraints**: Operating system must be a supported Android Version
- **Google Calendar Access**: The system may be limited on the access level of private Calendar events depending on previous OAuth
- **Hardware Level Constraints**: Hardware was support newer level Android operating systems, OS updates may depend on carrier / phone manufacturer.
- **Invalid Event Requests**: The system may be unable to create dynamically scheduled events if the user's calendar is already full to beyond reason.

5.0 Other Nonfunctional Requirements

5.1 Performance Requirements

None specified.

5.2 Safety Requirements

Not applicable.

5.3 Security Requirements

- Google Password Data shall not be accessible at any time in plaintext data
- When dealing with account data, all transactions will be handled using Google OAuth
- Calendar Events shall only be editable by the creator of the event

5.4 Software Quality Attributes

- **Stability**: The app shall remain free of crashes and data loss due to app misbehavior
- **Meets the Requirements**: The app shall maintain quality in terms of meeting the requirements specified in this SRS document
- **Modularity:** The app shall be modular in its design and create easily editable source code, using well-defined software design patterns when necessary.

5.5 Business Rules

None specified.

6.0 Other Requirements

None specified.

Appendix A: Glossary

The items listed in the glossary are technical terms used by this SRS or concepts/layouts specific to ATOMS.

- **Firebase Authentication**: ATOMS module specific to obtaining a google authenticated login from the user and establishing a connection to the Firebase platform.
- **Calendar Event View**: ATOMS module for grabbing calendar events from the Android Calendar Provider and displaying to the screen.
- **Add Events Module**: ATOMS module for adding events via user input and outputting to the Android Calendar Provider.
- **Edit Events Module**: ATOMS module for editing existing calendar events and outputting the changes to the Android Calendar Provider.
- **Settings Module**: ATOMS module for changing settings specific to the ATOMS application.
- **Settings Menu Options**: Menu Options in the Settings menu on the main app screen include "Settings" and "Log Out"
- **Settings Module Options**: Settings that can be changed include ATOMS color theme, Event time format, and Date format.
- **Auto Scheduled Calendar Event**: an event that is designated to be auto scheduled by the ATOMS application.
- **Statically Scheduled Calendar Event**: an event that is designated to be outputted to the Android Calendar Provider given the user's input by the ATOMS application.
- **Theme Listing**: Themes include dark mode, light mode, classic android, and oceanic.

Appendix B: Analysis Models

This project did not use any analysis models.

Appendix C: To Be Determined List

This project does not intend to use any new references.