
Software Requirements Specification

for

Schedify App

Version 1.0 approved

Prepared by

Kongkham Luangkhut

Bihaan Sapkota

Thompson Rivers University

November 13, 2024

Table of Contents

Table of Contents	ii
Revision History	iii
1. Introduction.....	1
1.1 Problem Definition.....	1
1.2 System Overview	1
1.3 Definitions, Acronyms, and Abbreviations.....	2
1.4 Assumptions and Dependencies.....	3
1.5 Purpose	3
2. Overall Description	4
2.1 Product Functions.....	4
3. Interface.....	6
3.1 User Interfaces.....	6
3.2 Hardware Interfaces	6
3.3 Software Interfaces.....	6
3.4 Communications Interfaces.....	6
3.5 Communications Interfaces.....	7
4. Functional Requirements	8
4.1 User Authentication Requirements	8
4.2 Automatic Schedule Extraction Requirements	8
4.3 Automatic Assignment Deadline Extraction Requirements.....	8
4.4 User Profile and Settings Requirements.....	9
4.5 Manual Schedule Management Requirements.....	9
5. Nonfunctional Requirements	10
5.1 Scalability and Future-Proofing Requirements	10
5.2 Performance and Efficiency Requirements.....	10
5.3 Security and Data Privacy Requirements.....	10
5.4 Usability and User Experience Requirements.....	10
5.5 Accessibility and Compatibility Requirements.....	11
5.6 Data Retention Requirements.....	11
6. Software Architecture	12
6.1 Client-Server Architecture	12
7. Release Plan	13
7.1 Phased Released Strategy	13
8. Training Plan.....	14
8.1 User Manual, and Documentation.....	14
9. Design Constraints.....	14
9.1 Platform, and Compatibility	14

9.2	Privacy and Data Security	14
9.3	Optimization for Mobile Devices	14
9.4	External Services Compatibility	15
Appendix A: Additional Recommendations		15
Appendix B: Contact information		15

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Problem Definition

University students frequently struggle with managing their academic calendars and assignment due dates. The idea to use Schedify in real life came about since the TRU app for students' schedules was no longer in use. Students could miss classes or deadlines since the current TRU system requires human scheduling. The Schedify App aims to expedite this process by obtaining schedules and assignments through interaction with Moodle and the TRU enrolling website.

1.2 System Overview

The Schedify App is an academic management and scheduling app for Android that has been created especially to assist college students in effectively managing their assignments and due dates. Schedify app is designed to streamline the process of managing academic schedules for students. Due to the fact that there is no proper mobile app that allows current TRU students to view their daily class schedule as well as assignment deadlines, they need to take a screenshot on registration page and check their every day class. Moreover, sometimes students might be too busy with their class schedule that they sometimes forget to submit that assignment. With this app, students can monitor all of their key academic reminders, deadlines, and upcoming classes anywhere anytime. This will save students time and lower the possibility that they will miss crucial deadlines by automating data retrieval from Moodle and TRU's registration page. All in all, This app seeks to improve academic efficiency and organization with features including calendar integration, notifications, and comprehensive course information.

1.3 Definitions, Acronyms, and Abbreviations

Term	Description
<i>Schedify App</i>	<i>Android-based scheduling tool designed to help university students manage schedules and deadlines.</i>
<i>Moodle</i>	<i>A Learning Management System (LMS) used by universities for course management and assignments.</i>
<i>Calendar View</i>	<i>A visual representation of schedules and deadlines in a calendar format, organized by date and time.</i>
<i>Timetable View</i>	<i>A section displaying the daily class schedule, including course names, timings, and professor details.</i>
<i>Assignment Deadline</i>	<i>The due date for course assignments that students need to complete and submit.</i>
<i>Notification Feature</i>	<i>Alerts users of upcoming deadlines, classes, and events through push notifications.</i>
<i>Google Drive Integration</i>	<i>A feature that allows users to back up assignment submissions and schedules to their Google Drive account.</i>
<i>API</i>	<i>Application Programming Interface used to connect the app with Moodle and other systems.</i>
<i>Sync</i>	<i>The process of retrieving and updating data from Moodle to the app.</i>
<i>User Profile</i>	<i>A section where users can manage their personal information and preferences within the app.</i>
<i>Manual Entry</i>	<i>A feature that allows users to add custom events or reminders to their schedule.</i>
<i>Offline Mode</i>	<i>Allows users to access previously synced schedules without an internet connection.</i>
<i>Authentication</i>	<i>The process of verifying a user's identity using their university credentials.</i>
<i>Client-Server Architecture</i>	<i>The design model where the Android app (client) communicates with a backend server to fetch data.</i>
<i>Google Drive Directory</i>	<i>A designated storage location on Google Drive where users can save copies of their submitted assignments and other coursework resources.</i>

Acronym	Description
SRS	Software Requirements Specification
LMS	Learning Management System
API	Application Programming Interface
TRU	Thompson Rivers University

1.4 Assumptions and Dependencies

The Schedify App relies on several assumptions to function correctly. First, it presumes that access to the Moodle API is permitted by the institution. Since the app pulls course schedules, due dates, and other academic information straight from Moodle, this access is essential. The app wouldn't be able to automatically sync the most recent data without access to this API.

Furthermore, the application makes the assumption that users have a reliable internet connection. For the app and the Moodle system to synchronize data in real time, internet access is necessary. The software does provide offline access to previously synced data, but in order to use features like scheduling updates and retrieving updated assignment details, an internet connection is necessary.

1.5 Purpose

This document serves as the Software Requirements Specification (SRS) for the Schedify App. It provides a comprehensive overview of the system's features, architecture, and development process. This document's goal is to assist the development team in accurately implementing the app's features while ensuring that it meets user needs. By defining clear requirements, this SRS aims to accelerate the project's design, development, and testing phases, ultimately leading to the successful deployment of the app for college students.

2. Overall Description

An outline of the main features, interface specifications, and system limitations of the Schedify App is given in this section. By connecting with the Moodle system to collect crucial information like class schedules and assignment due dates, the app is intended to simplify the process of organizing academic calendars for college students. The app's key functions include:

- *Automatic schedule extraction*
- *Assignment deadline notifications*
- *Calendar integration for viewing schedules*

2.1 Product Functions

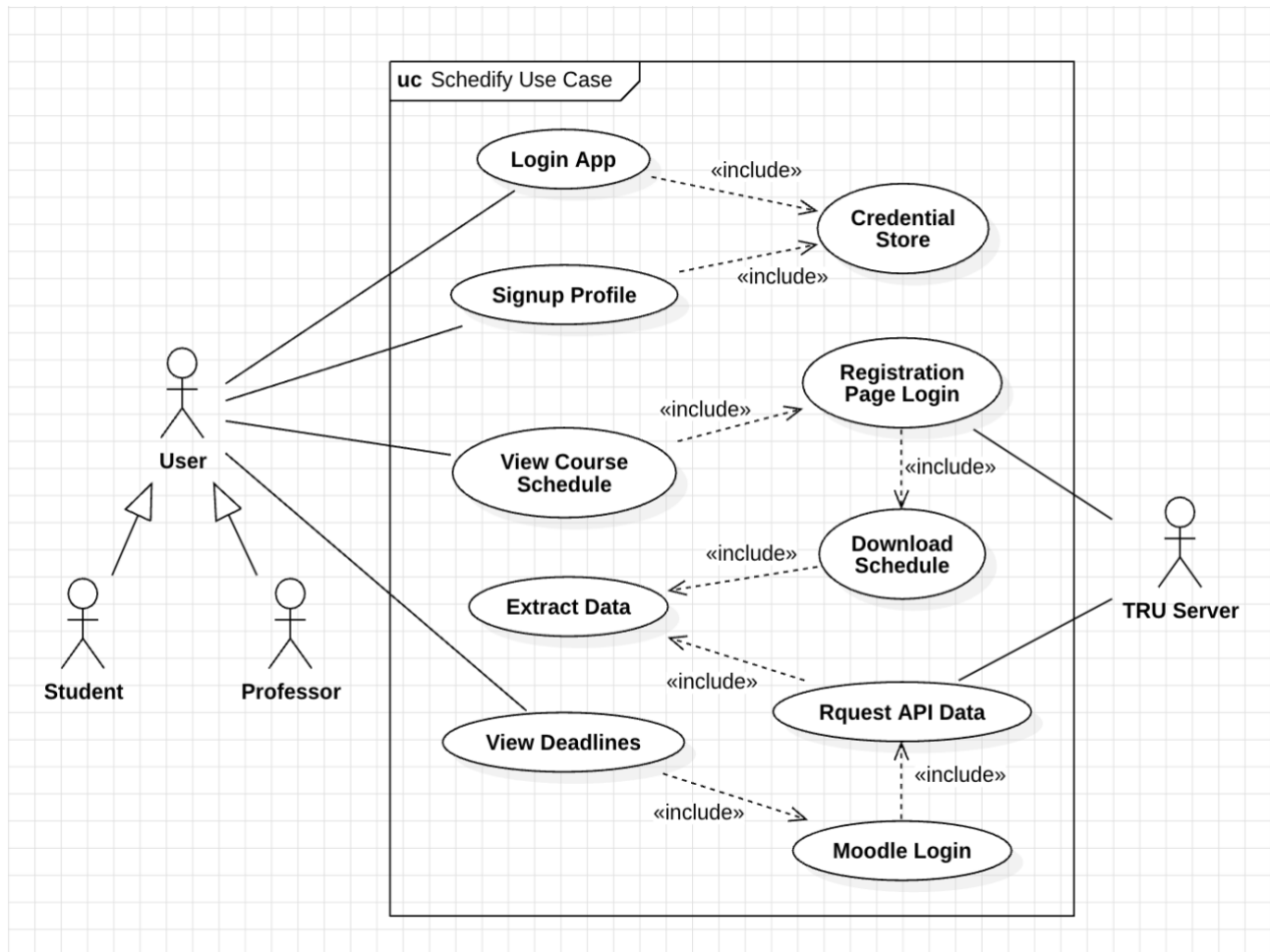
The Schedify App automates the process of fetching course schedules and assignment deadlines from the university's Moodle system. The main features include:

- *Automatic schedule extraction: The app connects to Moodle, retrieves students' schedules, and organizes them for easy viewing.*
- *Assignment deadline notifications: Users receive alerts for upcoming assignments, ensuring they stay on top of their coursework.*
- *Calendar integration: The app presents schedules in an integrated calendar format, allowing users to view their academic commitments at a glance.*

2.1.1 Use Case Model Survey

Schedify is intended to give students an easy-to-use experience. After logging in, users may examine their schedules, create reminders, personalize notifications, and access comprehensive course information. This feature guarantees that students don't miss any important deadlines and keeps them organized.

Use Case Diagram:



2.1.2 Actor Survey

The primary users of this app are mainly students who need a tool to manage their busy study schedules. Furthermore, academic advisors can also use this app to monitor students' progress and provide hands-on tasks.

3. Interface

3.1 User Interfaces

The Schedify App is designed with a simple and intuitive user interface. It includes dedicated sections on the bottom navigations for viewing schedules, create the tasks by teachers, profile, and adjusting user settings. The layout is optimized for ease of use, with clear navigation to help students quickly access the information they need.

3.2 Hardware Interfaces

Initially, the app is built for Android devices in Android Studio and requires a device running Android version 8.0 (Oreo) or higher. This ensures compatibility with most modern smartphones and tablets used by students.

3.3 Software Interfaces

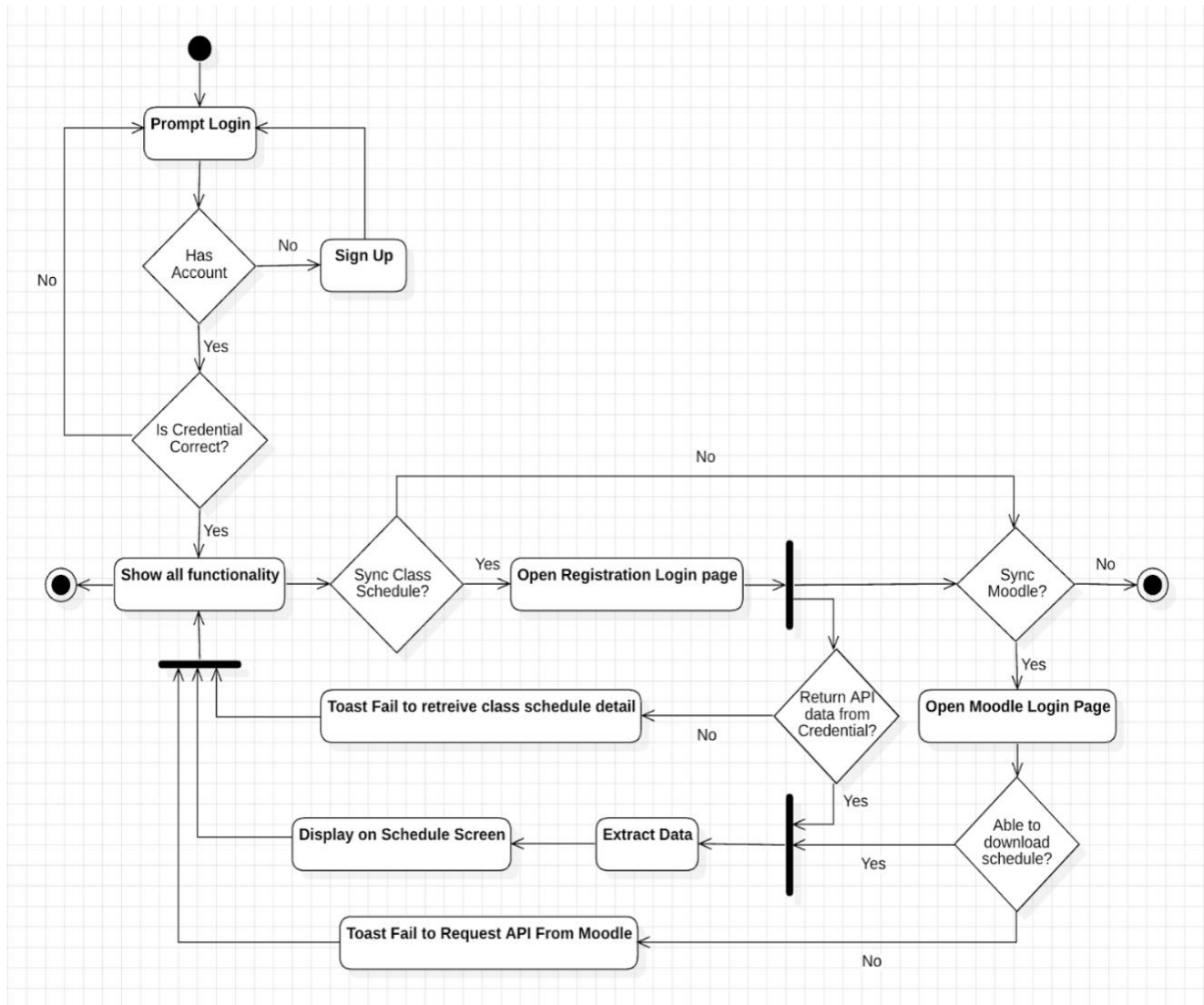
Schedify uses the Google Drive API to store assignments in the cloud and the Moodle API to store schedule and assignment data, allowing for easy integration with current systems. Data may be effectively retrieved, stored, and updated by the app thanks to this connection.

3.4 Communications Interfaces

The app uses secure communication protocols, specifically HTTPS, to protect data transmitted between the app (client) and its backend server. This ensures that all sensitive academic information is encrypted during transmission, maintaining user privacy and security.

3.5 Communications Interfaces

This activity diagram depicts the working model of the software. This allows client to see the visual/pictorial representation on how the software works



4. Functional Requirements

4.1 User Authentication Requirements

FR001 - The app must allow students to create a new account using their university email address and a secure password during the Sign-Up process.

FR002 - The app must support multiple users on the same device, enabling students to log in and switch accounts seamlessly.

FR003 - The app must provide the option to delete user profiles, allowing users to remove their accounts when no longer needed.

4.2 Automatic Schedule Extraction Requirements

FR004 - The app must connect to the university's portal and automatically retrieve course schedules using the Moodle API, organizing the data into a structured format.

FR005 - The app must display schedules in a day-by-day view, prioritizing courses based on time.

FR006 - The app must present schedules in an intuitive calendar format, allowing students to easily reference their daily, weekly, or monthly schedules.

FR007 - Once a scheduled class time has passed, the entry must automatically turn gray or disappear to indicate that it is no longer relevant.

FR008 - The app must allow users to click on a course entry to view additional details, such as the professor's name and course description.

FR009 - When a user clicks on a course's name, the app must redirect to the course page on Moodle for more detailed information.

4.3 Automatic Assignment Deadline Extraction Requirements

FR010 - The app must automatically sync assignment deadlines from Moodle at the end of each day using the API.

FR011 - Users must be able to view upcoming assignment deadlines and submit assignments directly within the app.

FR012 - The app must save all submitted assignments to a designated Google Drive folder, allowing users to access their historical submissions.

FR013 - The app must provide notifications for upcoming assignment deadlines to

ensure that students are reminded in advance.

FR014 - *Users must be able to customize the timing of assignment notifications, choosing how far in advance they receive alerts.*

FR015 - *The app must display assignments in order of due dates and prioritize them based on course importance, user-defined priorities, or keywords.*

4.4 User Profile and Settings Requirements

FR016 - *The app must provide a Profile Management feature that allows students to view and update their personal information.*

FR017 - *Users must be able to track courses they have already completed and review their progress.*

FR018 - *The app must include a Course Planner feature to help students plan future courses for upcoming semesters.*

FR019 - *The app must allow users to access and review all previously submitted assignments stored in the system.*

4.5 Manual Schedule Management Requirements

FR020 - *Users must have the ability to add, edit, and delete custom reminders and schedules within the app.*

FR021 - *New schedules added by users must automatically integrate into the existing deadline section and be sorted based on the specified due date.*

5. Nonfunctional Requirements

5.1 Scalability and Future-Proofing Requirements

NFR001 - The app must be designed to support future integration with additional student portals and databases without requiring significant reengineering.

NFR002 - The app must be capable of handling up to 10,000 active users simultaneously without degradation in performance, ensuring scalability as the user base grows.

5.2 Performance and Efficiency Requirements

NFR003 - Data synchronization with Moodle should occur only at the end of the day to minimize interruptions during peak usage times when students are actively working on assignments.

NFR004 - The app must load schedule data within 2 seconds of the user opening the app to provide a smooth and responsive user experience.

NFR005 - The app must be optimized for low power consumption to prevent excessive battery drain, ensuring it does not significantly affect the device's battery life during usage.

5.3 Security and Data Privacy Requirements

NFR006 - All user data, including profiles, schedules, and assignment details, must be securely encrypted to comply with university data privacy standards.

5.4 Usability and User Experience Requirements

NFR007 - The user interface must be intuitive and user-friendly, with clear navigation to schedules, assignments, and profile sections. Users should be able to access any information within three clicks.

NFR008 - The app's notifications system should be designed to be non-intrusive, allowing users to control notification settings and frequency to avoid unnecessary distractions.

5.5 Accessibility and Compatibility Requirements

NFR009 - *The app should support multiple languages to accommodate international users, with default options in English, French, and Spanish.*

NFR010 - *The app must be compatible with a wide range of Android devices, starting from Android version 8.0 (Oreo) and newer, to ensure accessibility for most users.*

5.6 Data Retention Requirements

NFR011 - *The app must store historical assignment submissions for up to two years, allowing users to easily access and review archived submissions.*

6. Software Architecture

6.1 Client-Server Architecture

6.1.1 Client-Side (Android Device)

- **Application Logic:** *The app is developed using Kotlin (or Java) and runs on Android devices. It handles user interactions, displays schedules, sends notifications, and retrieves data from the backend server.*
- **Local Storage (Caching):** *The app uses a local database (SQLite or Room) to cache important data like course schedules and assignment deadlines. This allows users to access their schedules even when offline.*
- **Google Drive Integration:** *To store assignment submissions, the app connects to the user's Google Drive account, providing cloud-based storage without needing a separate file server.*

6.1.2 Server-Side (Backend Server)

- **Database:** *The backend server uses a relational database (e.g., MySQL or PostgreSQL) to store user profiles, schedules, and assignment details. This database is synchronized with the Moodle API to keep data current.*
- **API Gateway:** *An API gateway manages communication between the app and Moodle. It retrieves academic data from Moodle and returns it in a structured JSON format to the Android app.*
- **Authentication and Security:** *The backend handles user authentication using secure protocols like OAuth 2.0. All user data is encrypted to ensure privacy and secure access.*
- **Push Notification Service:** *The backend integrates with Firebase Cloud Messaging (FCM) to send assignment reminders and notifications directly to the user's device.*

7. Release Plan

The release of the Schedify App will follow a phased approach:

7.1 Phased Released Strategy

7.1.1 Initial Release (Phase 1):

- *Once the contract is signed, development will commence.*
- *The first phase of the app will be released approximately 2 months after the contract sign-off.*
- *This phase will include core features such as schedule retrieval, assignment notifications, and calendar integration.*

7.1.2 Ongoing Development:

- *After the initial release, there will be monthly revisions to incorporate feedback from users.*
- *These revisions will focus on improving existing features, fixing bugs, and adding enhancements based on user feedback.*
- *Future updates may include additional functionalities, integration with other student systems, and performance optimizations.*

7.1.3 Subsequent Releases:

- *New versions will be released based on the feedback collected during monthly reviews and the development approach agreed upon (e.g., Agile or Waterfall).*
- *Each release will build upon the previous version, ensuring continuous improvement and feature expansion.*

8. Training Plan

8.1 User Manual, and Documentation

A comprehensive user manual will be provided to guide users on key features such as schedule setup, assignment management, and notifications. Additionally, in-app help screens and a downloadable PDF will be available to assist users. For more advanced support, such as for administrators, training sessions can be arranged upon request. Any additional training resources will be specified with associated costs upfront.

9. Design Constraints

9.1 Platform, and Compatibility

The Schedify App is specifically developed for the Android OS and is not compatible with iOS devices. The app requires Android 8.0 (Oreo) or later versions. It must consider differences in APIs and features across Android versions to ensure compatibility.

9.2 Privacy and Data Security

As the app handles sensitive academic data, such as student schedules and assignment deadlines, it must comply with data protection regulations (e.g., GDPR) if used globally. All data, including user credentials and schedules, must be securely encrypted.

9.3 Optimization for Mobile Devices

To minimize battery consumption, the app must efficiently manage background processes like data synchronization and notifications. This requires reducing the frequency of API calls and leveraging data caching to conserve device resources.

9.4 External Services Compatibility

The app relies on Moodle for data synchronization and Google Drive for storing assignment submissions. It is essential to maintain compatibility with these third-party APIs to ensure smooth data flow.

Appendix A: Additional Recommendations

We advise doing the following to get the most out of the Schedify App:

- 1. Integration of user input: After every release, collect user input on a regular basis to pinpoint areas that need work. This will guarantee that the application keeps changing in response to user demands.*
- 2. Constant Monitoring, and Updates: To fix security flaws, enhance functionality, and add new features, update the software frequently. Considering the dependence on third-party APIs like Moodle and Google Drive, this is particularly crucial.*
- 3. Scalability Considerations: To manage growing traffic, think about introducing load balancing and optimising server resources as the user base expands. This will support the preservation of app dependability and performance.*
- 4. Localisation Support: Adding support for more languages might increase the app's user base as it becomes more well-known, particularly in multilingual.*

Appendix B: Contact information

If you have any questions or require further assistance, please feel free to contact out:

- Contact: Kongkham Luangkhot, Bihaan Sapkota*
- Email: kongkham.luangkhot@gmail.com, bihaansapkota156@gmail.com*

We are always here to assist, and address the concerns you may have. Your feedback is invaluable for us to grow the Schedify App along with you.