

## Spaced Repetition - Design

Team 1

Sarah Reed: ree16014@byui.edu

Adrian Ruiz: adrianruizb18@gmail.com

Tom Metge: tom@metge.us

November 3, 2020

---

# PROJECT PROPOSAL

## Overview

We would like to work on an application that sends time-based reminders to help promote recall of information. The timing of these reminders is derived from research into spaced repetition. Information to recall is provided by the user and can be anything - text, image, link, audio, etc.

Our project sponsor is Virginia Metge, the 17-year-old daughter of Tom Metge. She would like to use the project to improve her memory as she studies her school work.

## Required Features

The minimum viable product will:

1. Collect information from the user:
  - Collect text-based information
  - Collect a text-based summary of the information (for use in the reminder)
2. Render reminders on a algorithmically-generated schedule:
  - Algorithm for time-spacing reminders
  - Display the user-provided summary for each item
  - Provide an interactive way for the user to view the full text or original information

## Stretch Goals

In addition to the MVP described above, we would like to:

1. Collection information from the user (in additional formats):
  - Collect scheduling or calendaring information
  - Collect photos
  - Collect PDFs
2. Render reminders:
  - Provide an interactive way for the user to view the additional formats via the reminder
  - Set a configurable timer during which the reminder remains on screen (so the user can't dismiss and ignore it)
3. Syncing data across devices:
  - Use a data API to store and sync collected information and reminder schedules across multiple devices.

## Approach

---

---

## **PERSISTENCE**

For local and cloud storage, we will use Firebase.

## **AUTHENTICATION**

To identify and authenticate the user, we will integrate the stock Android login flow (via Android Studio) with Firebase.

## **REMINDERS**

Firebase provides support for local and remote notifications. We will integrate both to provide a seamless user experience (see user cases).

---

## USER STORIES

### **Primary Use Case (adding and consuming a text-based reminder)**

Virginia (our sponsor) is struggling to recall information necessary to succeed in several classes in school. She is specifically concerned with Math. She opens the Spaced Repetition app and:

1. Presses the "+" button to add a new tidbit for recall
2. Fills in the text summary with: "Area of a circle"
3. Fills in the text to recall with: "Area =  $\text{Pi} * (\text{Radius} * \text{Radius})$ "
4. She taps save and the new tidbit is added to her list
5. She closes the app and continues with her day
6. In the evening, the app presents a reminder with "Area of a circle" included
7. Virginia attempts to recall the equation and taps the button in the reminder to check
8. The app presents the text to recall ("Area =  $\text{Pi} * (\text{Radius} * \text{Radius})$ ").
9. Virginia again closes the app
10. The following morning, the app presents a reminder with "Area of a circle" included
11. This sequence is repeated two days later, then 4 days, and again one week later

### **Primary Use Case (specifying a termination date for a reminder)**

1. Steps 1-3 of the above case
2. She taps on the option to end the reminder on a fixed or relative date.
3. Steps 4-11
4. On the date of the reminder's termination, the app stops presenting the reminder.

### **Primary Use Case (removing an existing reminder)**

1. She selects a reminder from the list of existing reminders
  2. The app shows a view asking her to edit or remove the selected reminder.
  3. She select "Remove"
  4. The reminder is removed.
-

---

### **Primary Use Case (editing an existing reminder)**

1. She selects a reminder from the list of existing reminders
2. The app shows a view asking her to edit or remove the selected reminder.
3. She selects "Edit"
4. The app presents the same interface as for adding a reminder, pre-populated with the data from the existing reminder.
5. She taps "Save"

### **Stretch Goal Case (adding and consuming non-text reminders)**

As a stretch goal, we would like the app to accept additional formats of information. The user story for this case is as follows:

1. Virginia presses the "+" button to add a new tidbit for recall
  2. She fills in the text summary with: "U.S. population in the 20th century"
  3. She taps "Add photo" and adds a JPG with a chart showing the population of the U.S. throughout the 20th century
  4. She taps save and the new tidbit is added to her list
  5. The app offers reminders as described in the first story (the base case)
-

# INTERFACE

## Main Screen



---

## Add or Edit

Edit Reminder

Summary: Area of a circle

Contents:

$$A = \pi * r^2$$

This is the area of a circle... (and any other notes the user wants to add).

Days to remind me: 30

☒

---

Add or Edit Pop-up

Area of a circle...

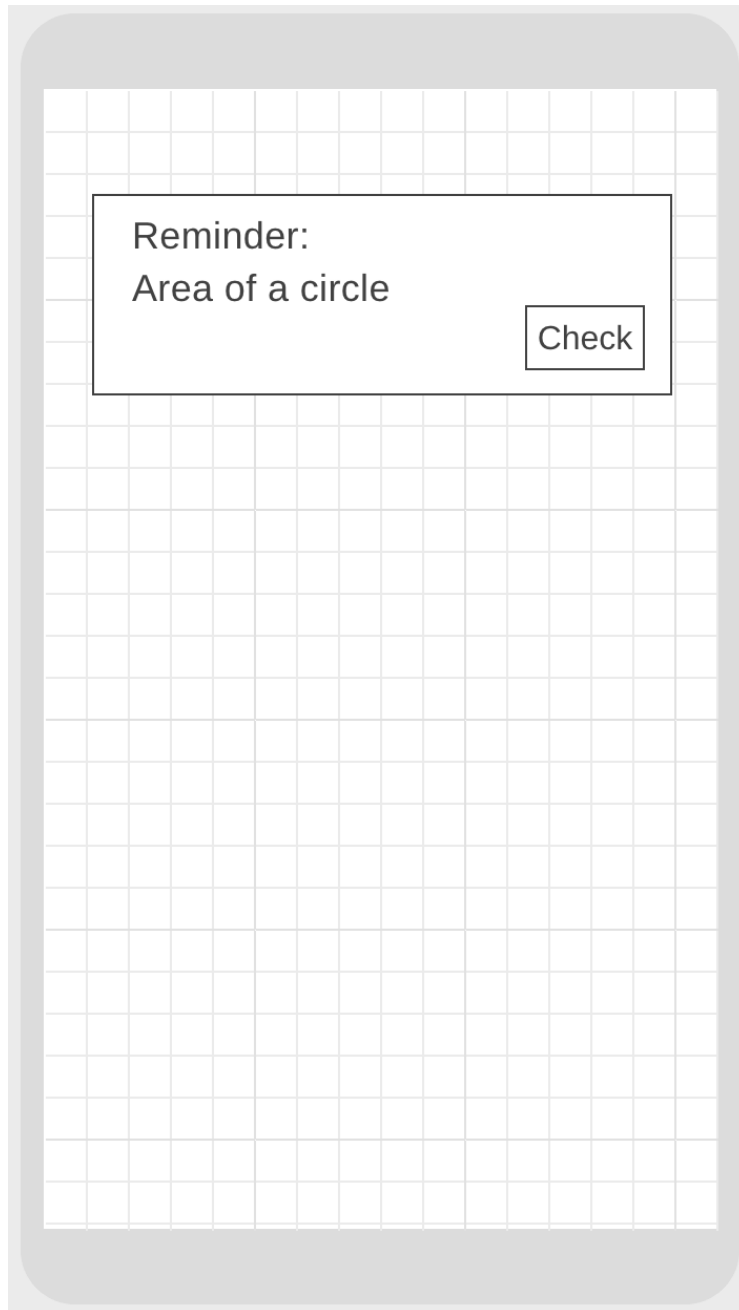
Edit

Remove



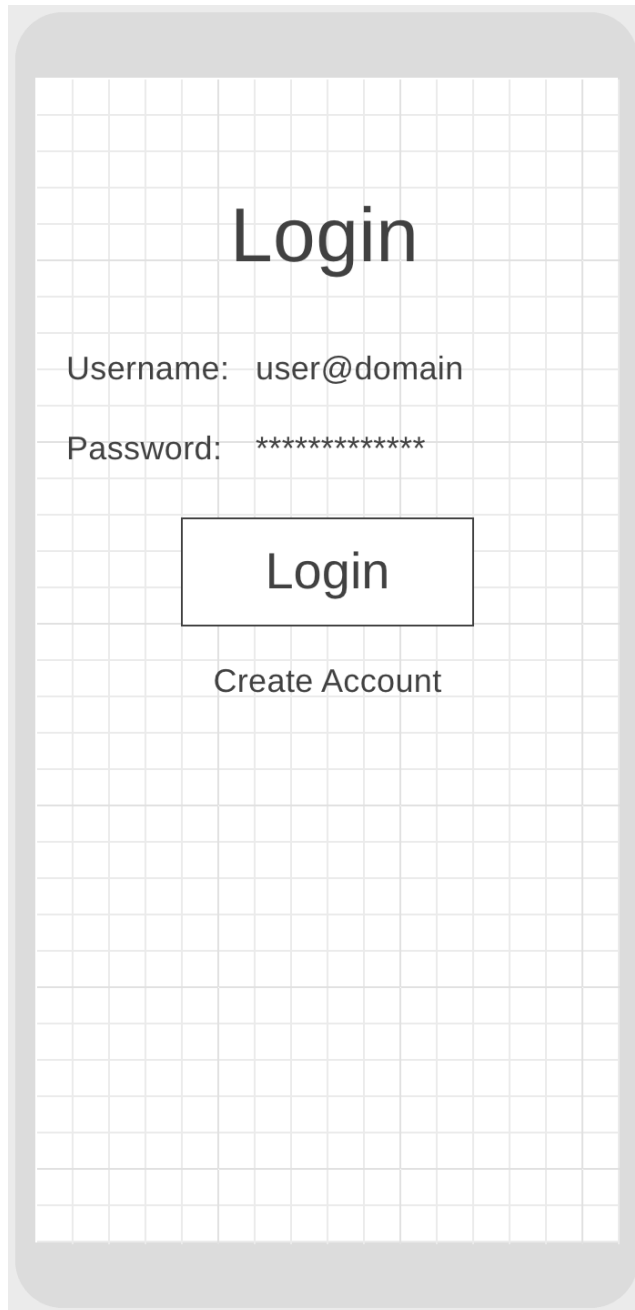
---

## Reminder



---

## Login

A login form mockup displayed on a light gray grid background, enclosed within a rounded rectangular frame. The form contains the title "Login", a "Username:" label followed by the text "user@domain", a "Password:" label followed by ten asterisks, a "Login" button, and a "Create Account" link.

Login

Username: user@domain

Password: \*\*\*\*\*

Login

Create Account

---

---

## Create Account

# Create Account

Username: virginia@metge.us

First name: Virginia

Last name: Metge

Password: \*\*\*\*\*

Cancel



## DATA MODEL

### Reminder

- User
- Summary
- Content
  - Text
  - Image
  - Document
- Creation time
- Time to live (TTL)
- Last notification date
- Current period

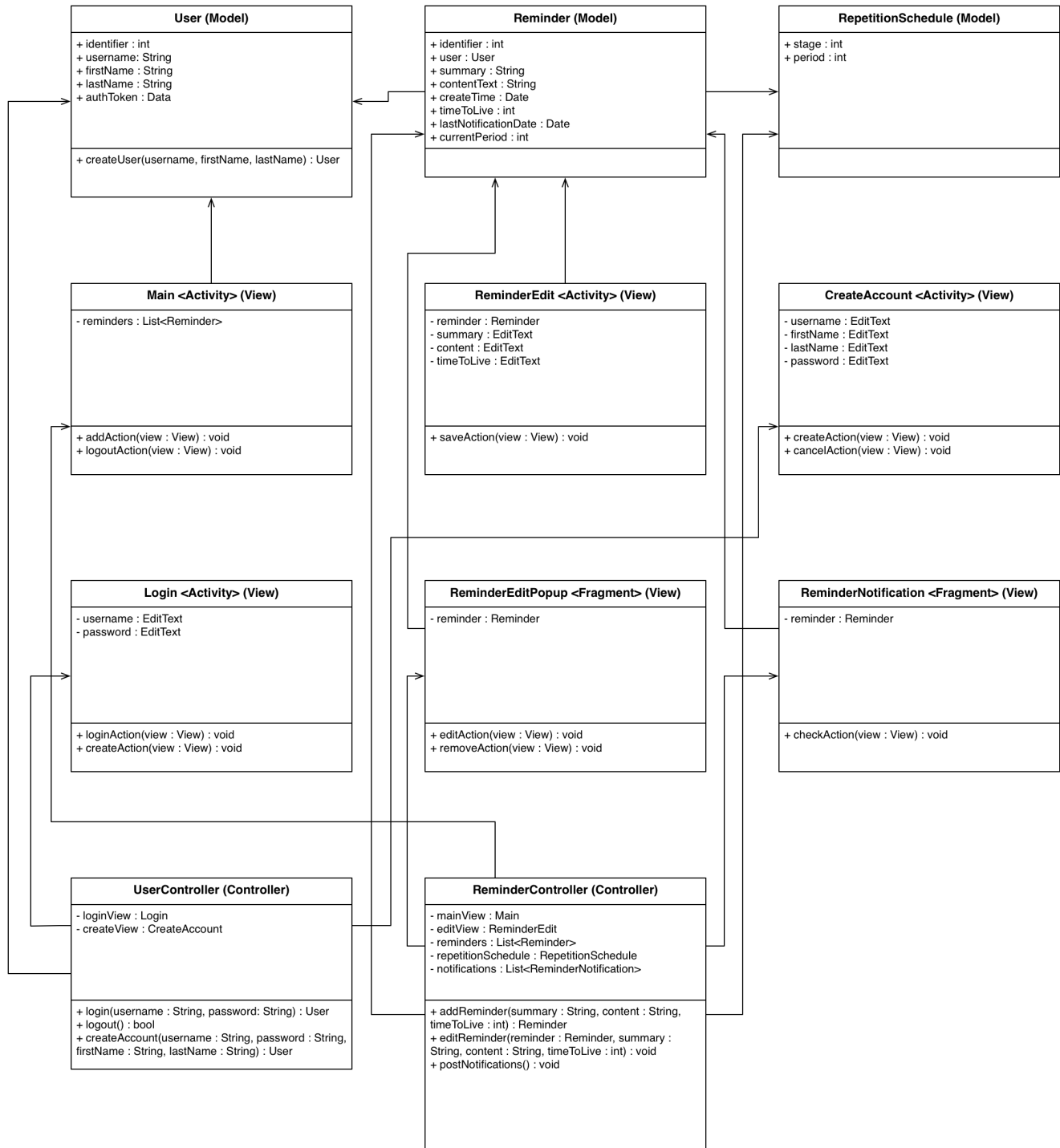
### User

- Username
- First name
- Last name
- Auth token

### RepetitionSchedule

- Every day for 7 days
  - Every week for 4 weeks
  - Every month perpetually (until TTL)
-

## UML



## MILESTONES

### **Milestone 1 (Lesson 09)**

- Create all views with all visual elements present
- Use mock data (not live data)
- Main view and associated controls are fully functional (with mock data)
- Initial feedback from stakeholder

### **Milestone 2 (Lesson 11)**

- Integrate live data
- Add unit test coverage for data model
- All views and controls are fully functional
- Follow-up feedback from stakeholder

### **Milestone 3 (Lesson 13)**

- Test on different devices and device classes
  - Final testing and feedback from stakeholder
-