How to Use this Template

1. Make a copy [Select All → Copy → Paste into new document]
2. Name your document file: "**Capstone_Stage1**"
3. Replace the text in green

GitHub Username: petemit

Time Goalie

Brief Google Play/like Description

You are the Time Goalie! Clear all those goals before they reach your net!

Meet your goals and save your time! Track the goals you want to do more of, like exercising, and track the goals you want to limit, like naptime!

Time Goalie is a habit tracker that allows you to set Time Goals, of which you can configure timers that will notify you when your allotted time for a goal is expired.

Examples:

- Set a 1 hour a day running goal. You CLEAR your goal when you run at least an hour!
- Limit your time browsing Facebook by setting a 1 hour a day goal. You CLEAR your goal if you spend less than 1 hour looking at cat videos on Facebook.

Additional features:

- Keep track of simple yes/no habits in addition to Time Goals
- Configure notifications at the end of your time goals

Full Description

Time management apps are prolific on the Google Play store! If you want to make a daily schedule or if you want to track habits, then there are hundreds of free options already available. Reaching our goals can be helped by these apps, but sometimes we want to limit our time spent on our goals.

Problem:

The user has three types of goals: time goals they want to limit (such as playing video games), and time goals they want to encourage (such as learning to juggle), and simple yes/no goals (such as brushing your teeth). The user wants a way to balance the time spent on their time goals, whether excess time is acceptable or is less acceptable.

Proposed Solution:

An app that has three types of goals you can set: Time goals where excess time is desirable, time goals where excess time is less desirable, and simple yes/no goals. The app will present the days goals in a single page so the user can track their progress.

The time goals will be set up with timers with time limits the user wants to set.

The differentiating component of the app is that the user can start a goal, and a timer will be started tracking the user's progress. At the end of the time limit, the user will receive a notification that the time limit has expired, and by default, will continue to notify the user at a defined reminder interval.

Example 1:

- The user wants to set a time goal of practicing to juggle for 20 minutes a day. The user will create a new goal called "practice juggling". The goal will now populate the main screen with a progress bar.
- The user, when ready to practice juggling, will press the start button for that goal, and then the timer will start, and the progress bar will begin to fill up.
- (configurable by the user) Once the time has elapsed, the app will notify the user, and the GOALS CLEARED counter will increase by 1.
- (configurable by the user) If the user does not stop the time goal, then it will continue to notify the user at a configurable interval.

Example 2:

- The user wants to set a time goal of limiting naptime to 20 minutes a day. The user will create a new goal called "20 minute naptime". The goal will now populate the main screen with a reversed progress bar.
- The GOALS CLEARED counter will be increased by 1.
- The user, when they start their nap, will press the start button for that goal, and then the timer will start, and the progress bar will decrease.
- (configurable by the user) Once the time has elapsed, the app will notify the user--they will have a minute to stop the timer. If the user does not stop the timer, then the GOALS CLEARED counter will decrease by 1.
- (configurable by the user) If the user does not stop the time goal, then it will continue to notify the user at a configurable interval.

Example 3:

- The user wants to set a yes/no goal for watering the plants. The user will create a new goal called "water plants". The goal will now populate the main screen with a checkbox.
- The user, when they finish the task, will check the box.
- The GOALS CLEARED counter will be increased by 1.

Now, the user can track the time spent on their goals and additionally is assisted in balancing their time.

Intended User

People who want a simple habit tracker, but also want to track how much time they spend on their habits, both desirable or not desirable.

Features

- Saves information
- Receives notifications
- Receives reminders
- Uses timers
- Picks moderately complex date and time arrangements
- Allows user preferences to customize
- Shows Reports

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

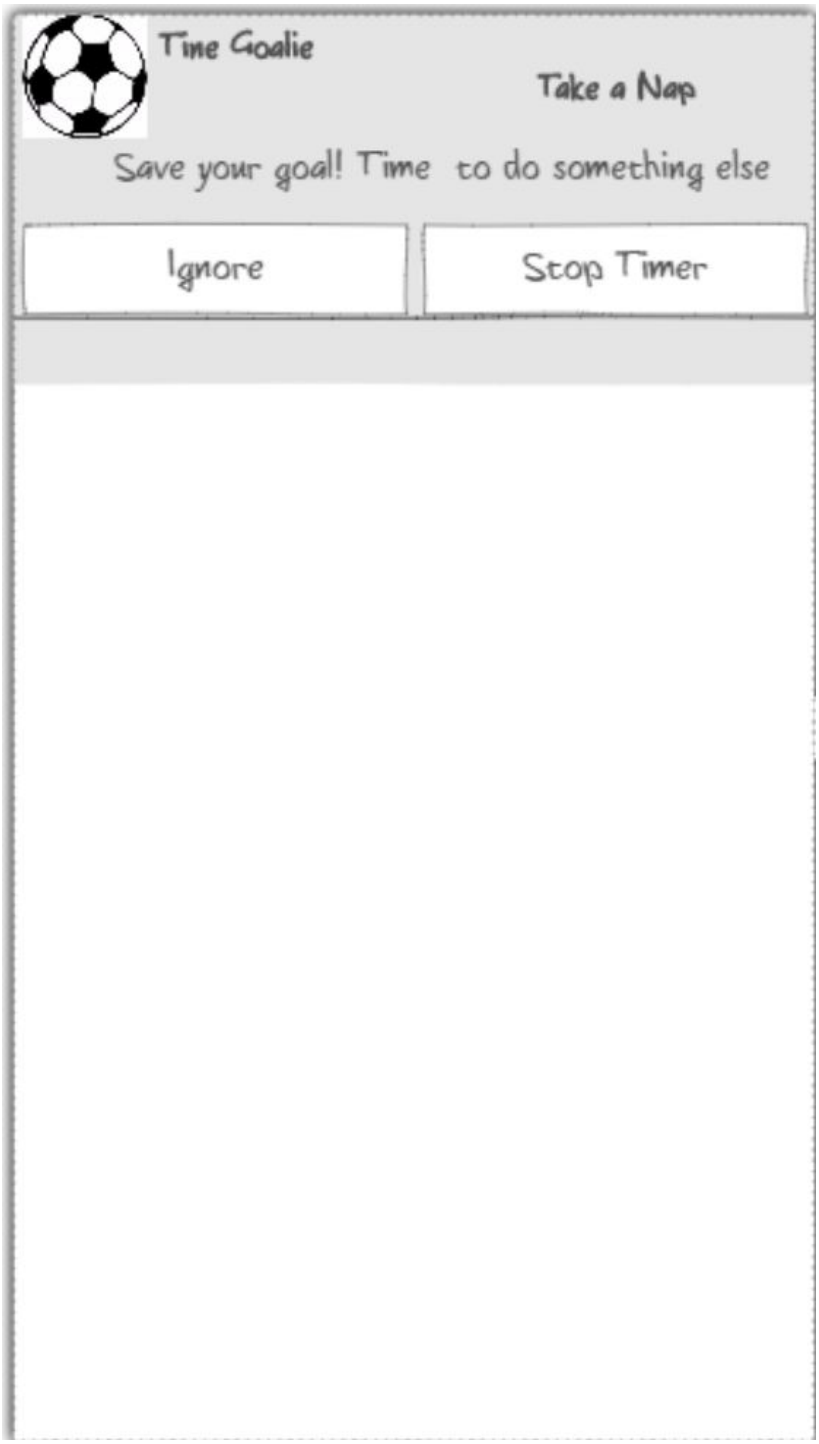
Screen 1

Main Launcher brings user to main goal view.
User can see all current goals in a scrolling view, and can swipe left or right to see specific days.

As timers start, the soccer ball progresses to the right. If the goal is a goal that the user wants to encourage more time spent, then the goal will be on the left, and the ideal is to get the soccer ball as far away as possible.
If the goal is a goal that the user wants to limit, then the goal will be on the right, and the user is motivated to prevent the soccer ball from entering the goal.

User is currently taking a nap, and will be notified when timer expires.

Screen 2



User ignores notification

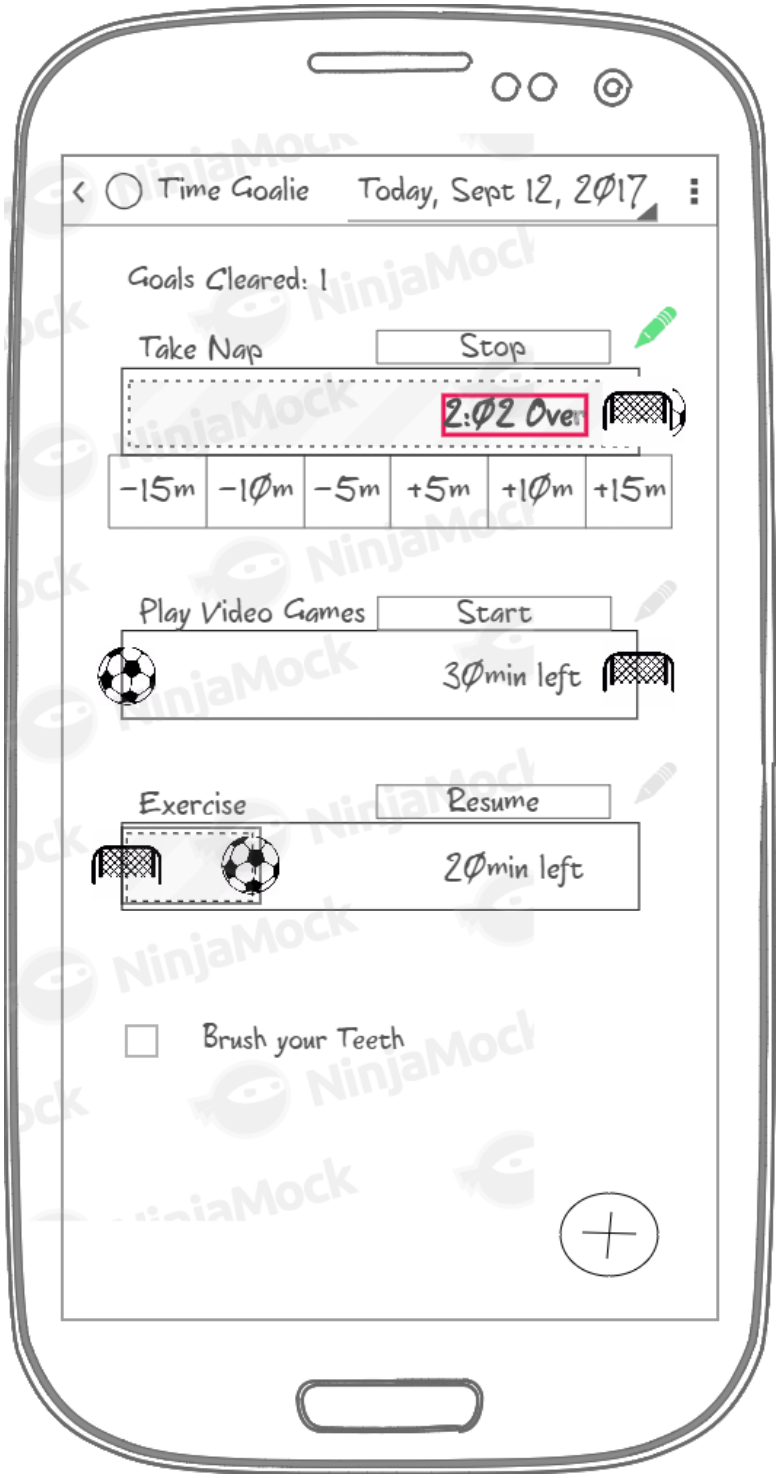
Screen 3

User wakes up and sees their widget, and clicks Time Goal to return to the app.

Screen 4

Goals cleared counter decrements by 1, and user decides to tap the pencil icon to see the edit options.

Screen 5

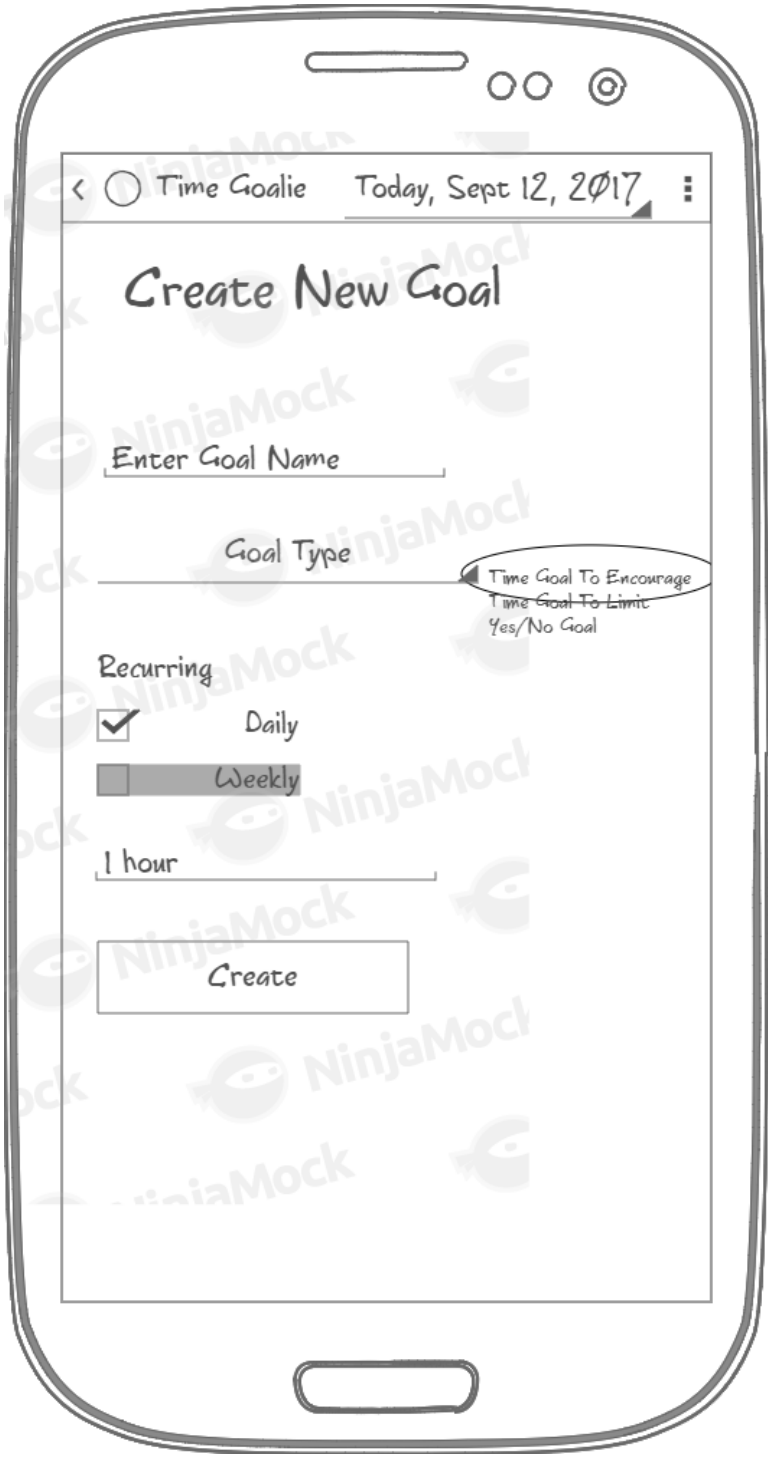


User observes the editing options and then decides to stop the Take Nap timer and resume the exercise timer.

Screen 6

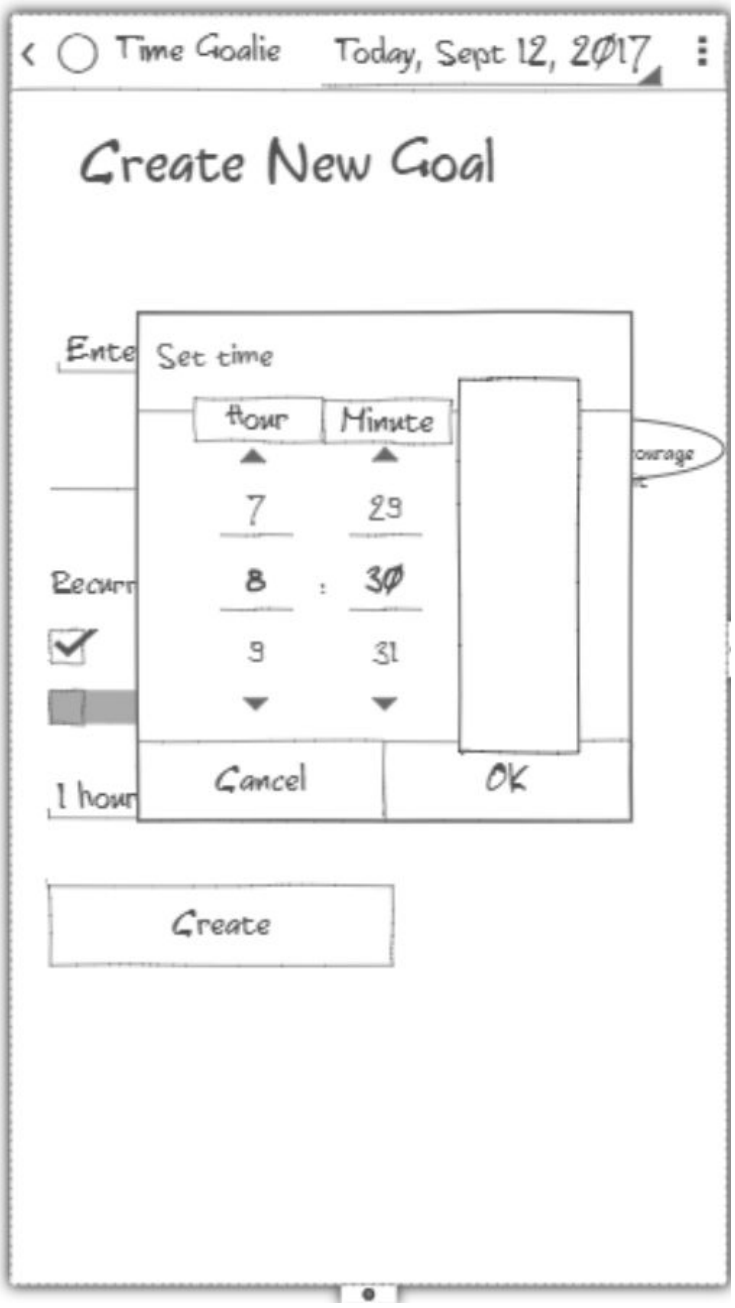
User comes back to the app after exercising and gets the success dialog. User wants to create a new goal, and clicks the + FAB.

Screen 7



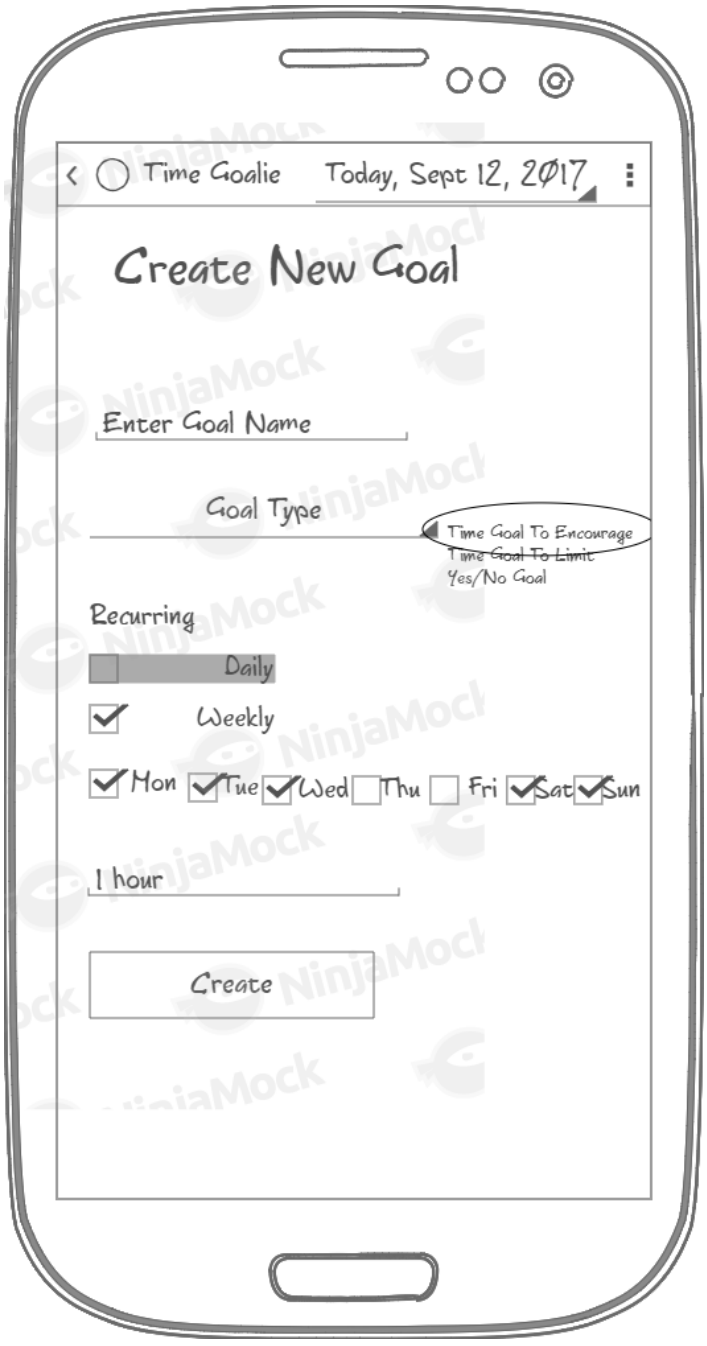
User selects “Time Goal To Encourage” and checks the daily box, and clicks the time picker to see what the time selection is like.

Screen 8



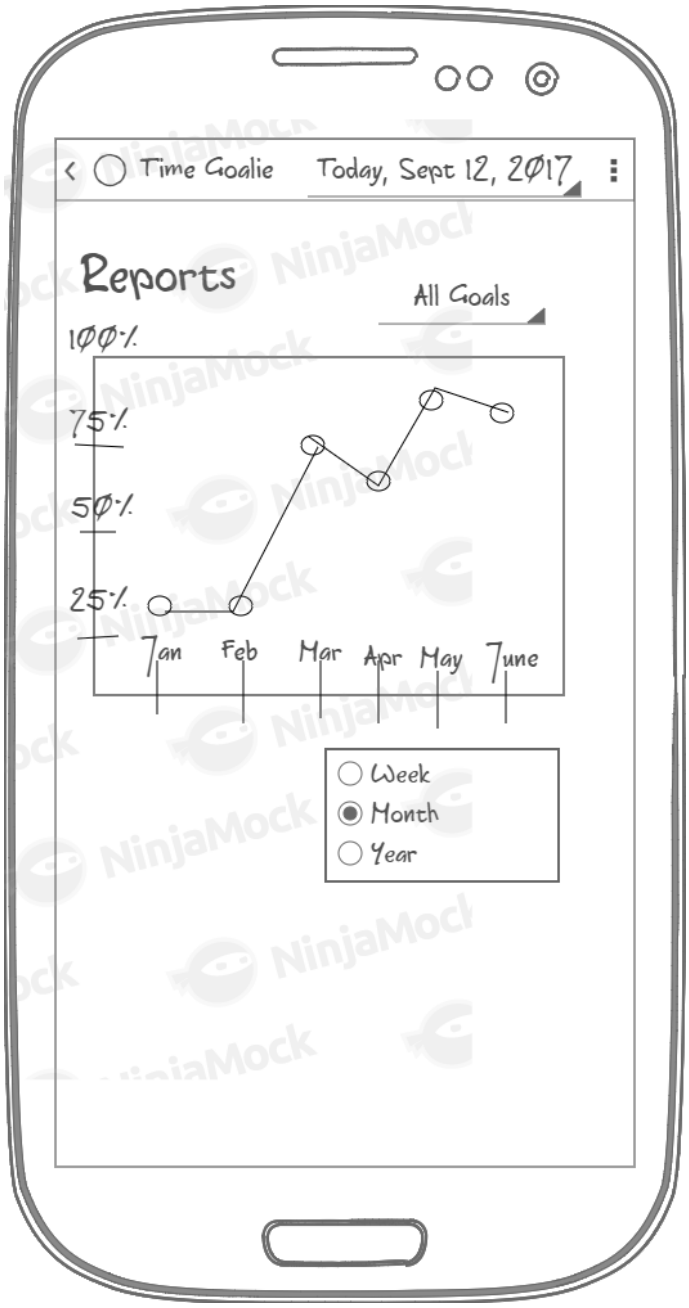
User flips through the time selection options, and clicks cancel. User then clears the daily checkbox and clicks the weekly checkbox.

Screen 9



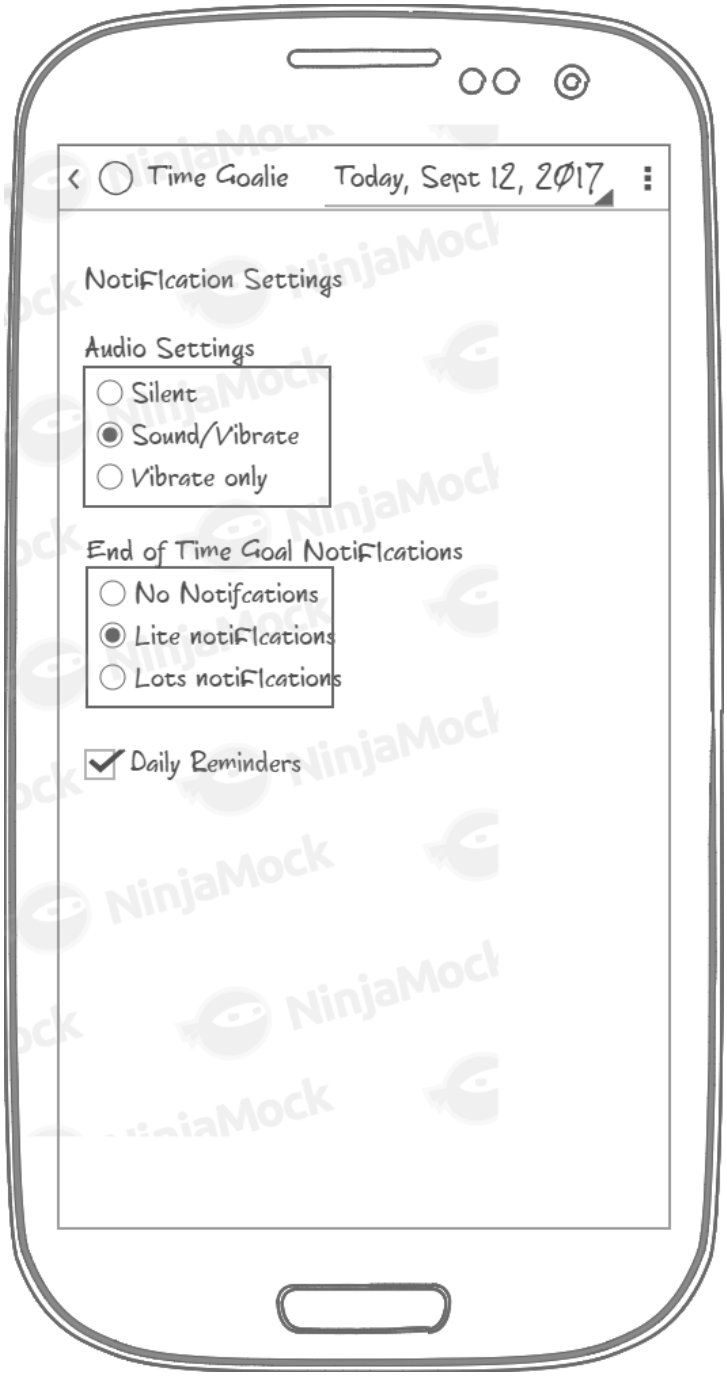
User then explores the weekly recurrence checkboxes, and then clicks the menu button and selects Reports.

Screen 10



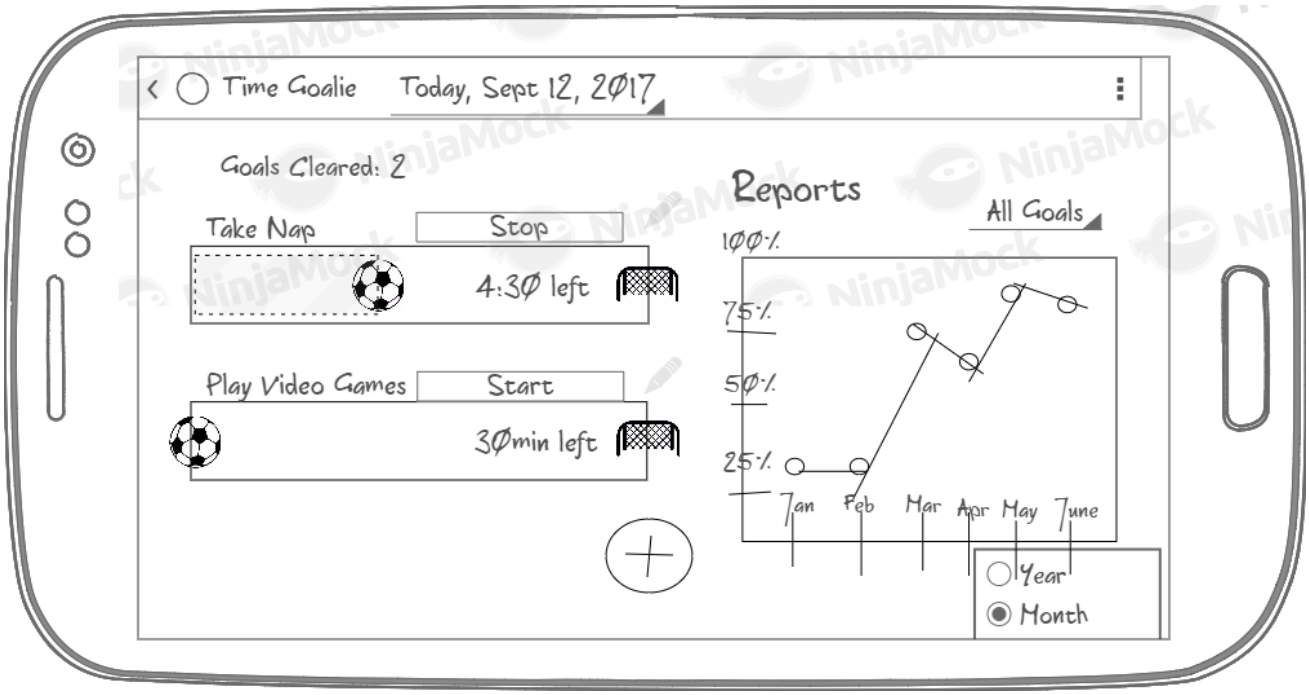
User selects through the week, month, and year selectors to examine their progress.
Additionally, they select different goals so they can see the individual percentages for each goal.
User clicks the Settings option in the menu.

Screen 11



User explores the settings on the preferences screen. User then purchases a tablet and then goes to the main screen to see how their favorite app looks

Screen 12



User's tablet, which looks a lot like a big phone, splits the main view into a dual pane view that shows the reports.

Key Considerations

How will your app handle data persistence?

The app will handle data persistence through a local SQLite database. The app will use a custom content provider with a SQLite contract to get access to the DB. Data will be loaded to the UI using a loader.

Describe any edge or corner cases in the UX.

Since all screens are only 1 screen away from the main goal screen via the menu, the up button will navigate back to the main screen. The back button will return to the previous function. For example, if you are creating a new goal and you click on the settings option from the menu, and then click the back button, the back button will return you to the create new goal screen, while the up button will take you back to the main screen.

Describe any libraries you'll be using and share your reasoning for including them.

*I will be using Firebase Analytics for the purpose of finding out usage information about the app.
*I will also be using MPAndroidChart, from Phillip: https://android-arsenal.com/details/1/741
MPAndroidChart will help me in implementing the reports page.

Describe how you will implement Google Play Services or other external services.

I will be using Firebase Cloud Messaging so as to allow push notifications for daily reminders (to save on battery).
I will be using Firebase Analytics to capture data from the app usage, in the hopes to provide shortcuts for future users on what goals to make.

Next Steps: Required Tasks

Task 1: Project Setup

1. Configure new project with the dependencies set up for FCM, Firebase Analytics, and MPAndroidChart.
2. Sync up app with Firebase account
3. Find the graphics I need, such as the vector soccer ball, and a vector goalie net.
4. Decide on a font that fits the theme of the app.
5. Decide on a color theme

Task 2: Implement UI for Each Activity and Fragment

- Build empty main activity to be able to use fragments to support multi-pane layouts
- Build the main goal fragment UI.
- Construct the UI for the reports fragment and get the MPAndroidChart view set up
- Construct the preferences screen
- Construct the Create Goal Screen
- Construct the widget UI
- Construct the notification UI
- Make a layout that will support the reports and the main goal fragment in the same landscape tablet view.

Task 3: Build the Content Provider

Draw out the relationships for the local SQLite DB, and create a contract and content provider to go along with this data.

- By default as I'm testing, when I reinstall the app, delete and rebuild the database
- Design DB
- Create Contract
- Content Provider

Task 4: Build logic for Main Goal Screen

Create goal objects that will interact with dates and times, as well as allow for tracking its goal information, and have the data necessary to start and stop timers.

- Use a Loader to load data from sqllite db into the views.
- Create Test Goal objects that can be instantiated that will help with testing.
- Build timer service that will allow for timers in the background even when the app is closed
- Build logic so that 3 goal types correspond to the 3 different UI layouts.
- Allow editing of the timer time with the + or - fields as seen in the mock UI.

Task 5: Create New Goal logic

Need to build out the New Goal logic to allow for custom goals.

- Implement input validation
- Construct recurrence picker
- Construct time picker to select how many hours (max of 24)

Task 6: Create Notification system, and hook into Firebase

Create a broadcastreceiver for the notification system and implement notifications for end of timer, as well as daily reminders through Firebase.

- Allow customization of these reminders through preferences
- Ensure you can stop the timer from the notifications, return to the app, or ignore the notification

Task 7: Create logic for Reports screen

Create Reports screen with MPAndroidChart that allows for showing percentages for day, month, and year filters, and you can also filter by a singular goal, or all of them.

- Ensure that this reports screen looks correct in multi-pane layout as well as single-pane layout
- Reports must be gathered via Async tasks

Task 8: Create Widget

Create broadcastreceiver for the widget and allow for tracking of active goals.

- Allow at LEAST stopping of active goals, but if you are feeling ambitious, a bonus of showing all of today's goals with the ability to START them as well.

Task 9: Create logic for preference screen

Allow customization of notifications, including audio settings, here.

Task 10: Polish

Ensure everything looks good, all strings/dimens are extracted, any and finally, allow for updating of database.

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"