How to Use this Template
1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions
1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: fruitspunchs

Productivity Up

Description

Why do we want to be more productive?

Do we want to excel at work?

Or finish work to do the things we love?

Based on The Science of Productivity and Sparring Mind, Productivity Up is a set of tools based on scientific research to help increase your productivity.

Problem:

The user wants to apply productivity techniques suggested by The Science of Productivity but using multiple separate applications incurs a lot overhead such as searching for apps and switching between apps.

Solution:

Design an app implementing all productivity techniques suggested by The Science of Productivity. Implement a home screen to present relevant and timely information for each technique, and implement a detail screen to focus on each technique.

Intended User

Students, Professionals, Businessmen, Hobbyists

Features

- Overview - gives you information of all productivity tools at a glance
- Pomodoro timer - stay refreshed by dividing work into 25 minutes with breaks
- Accountability chart - keep track of what you actually do during work time
- Ultradian rhythm meter - work and rest with your natural rhythm
- Action plan - list your top 5 tasks for today and tomorrow
- Calendar deadlines - reminds you of your important deadlines
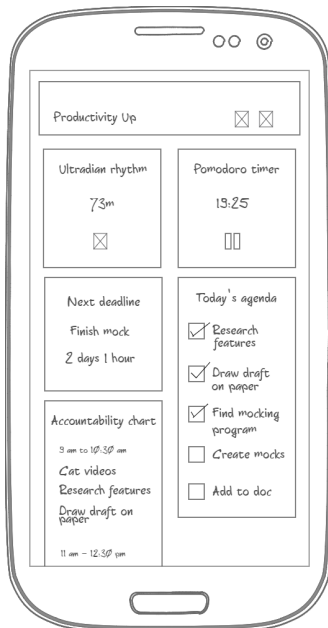
User Interface Mocks

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

Home Screen (No card titles)
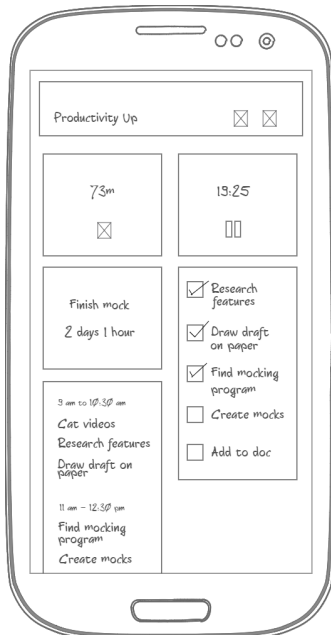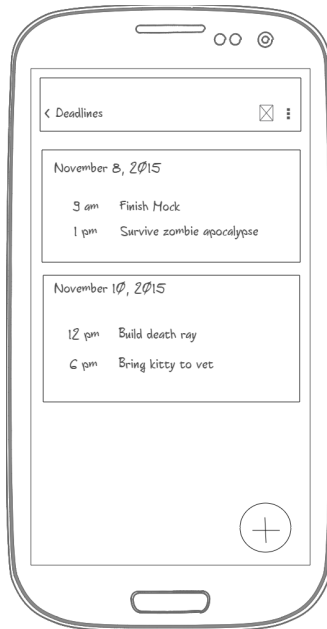

This is where you can access all the productivity tools. The action buttons enable card titles and share the app.

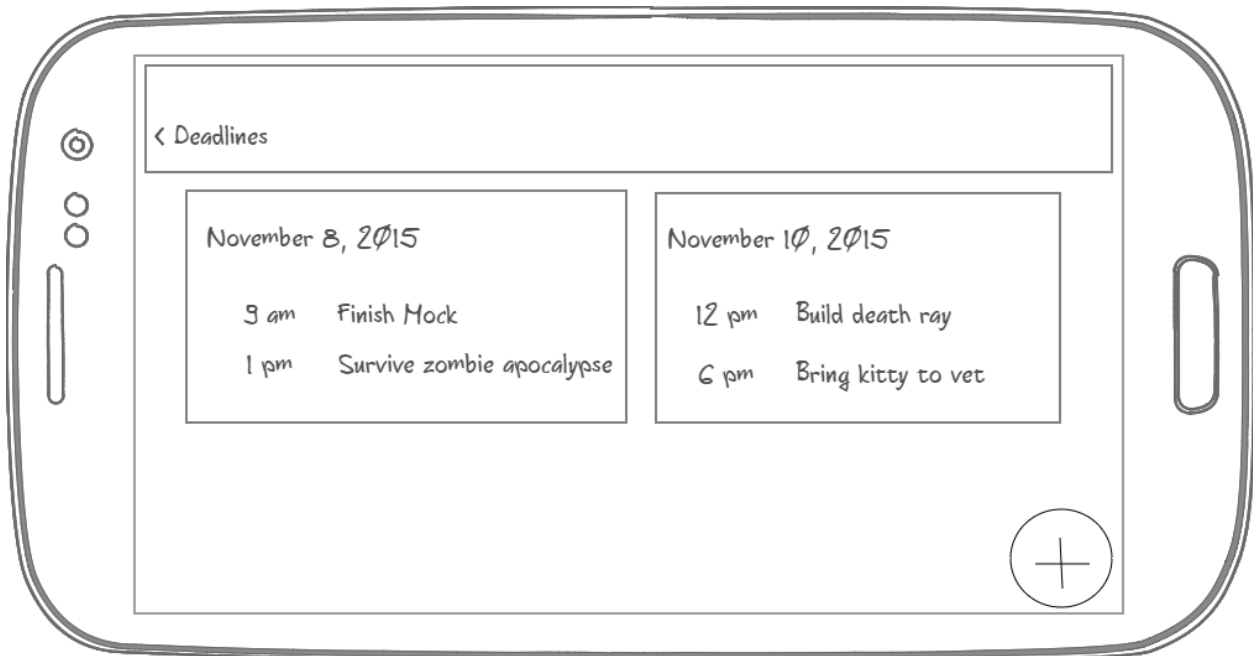## Home Screen (Card Titles)

Alternate home screen with card titles enabled. Card titles are enabled by default.

Deadlines

Daily view of deadline reminders. Add deadlines using fab.

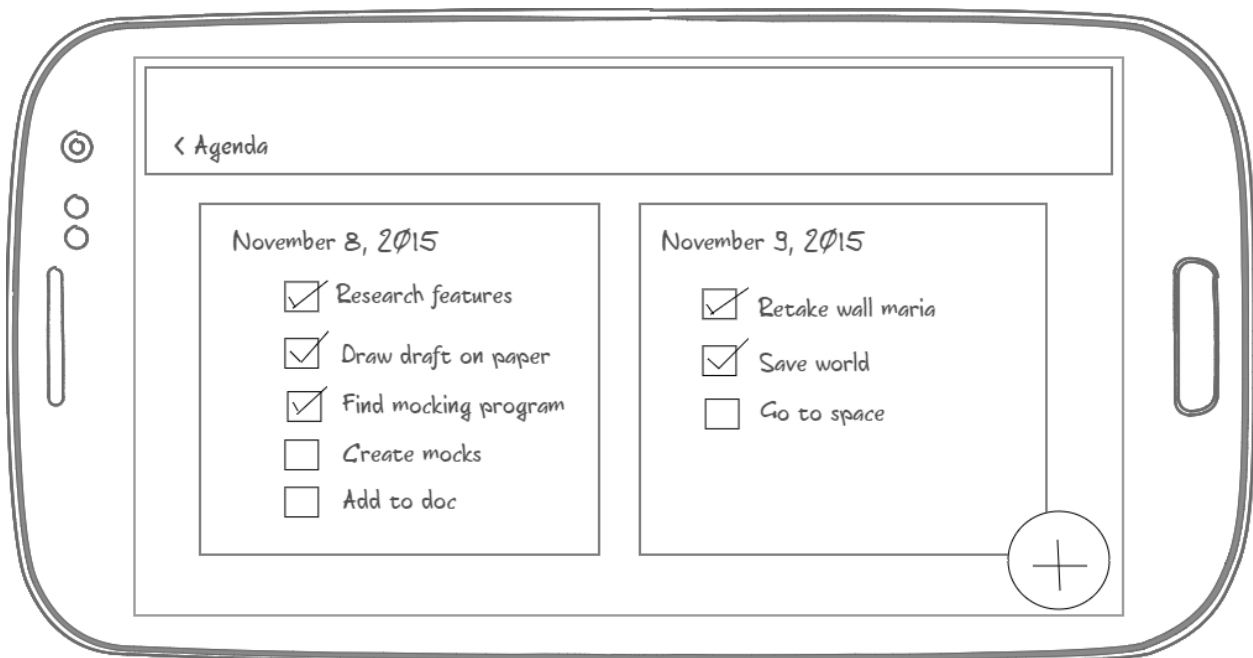## Deadlines (Tablet)

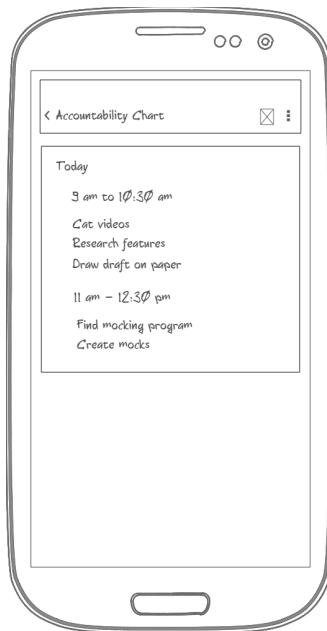Tablet view of deadline reminders.

Agenda
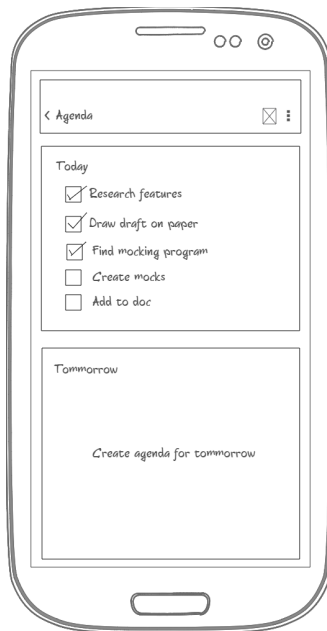
List tasks to be done for today and reminds to add tasks tomorrow. Top 5 tasks recommended.
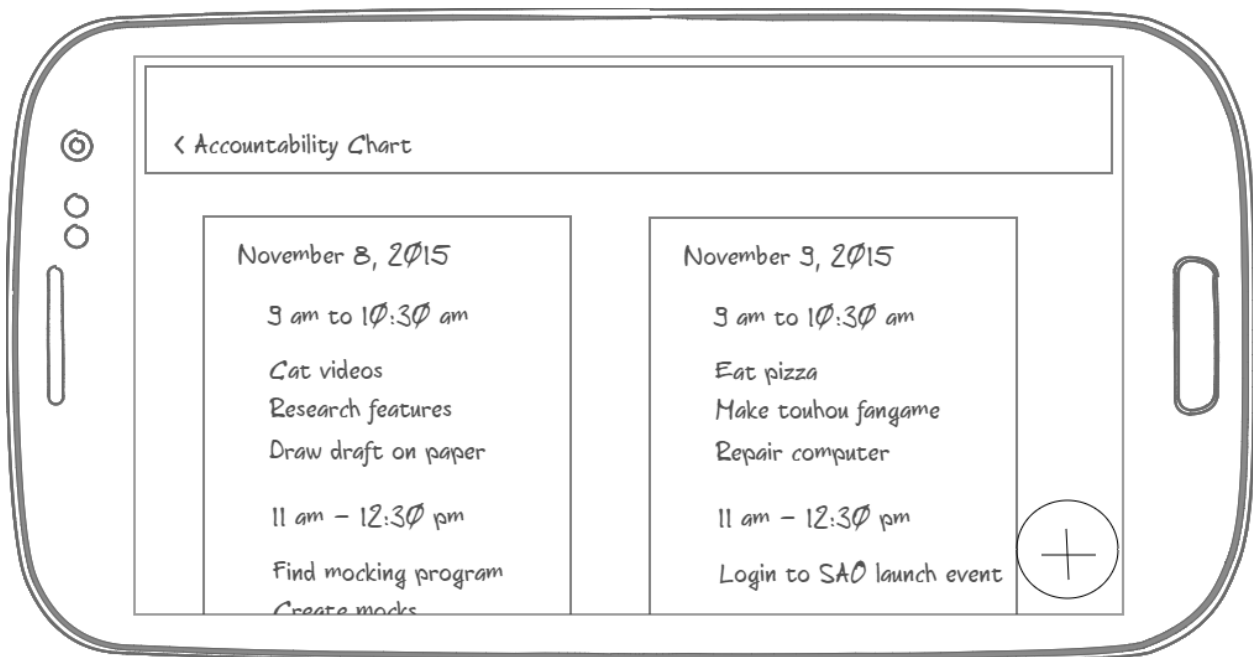
Agenda (Tablet)

Tablet view of agenda.

Accountability Chart

Records how you spend your time every 90 minutes followed by an unrecorded 30 minute break.

Accountability Chart (Tablet)

Tablet view of accountability chart.

Key Considerations

How will your app handle data persistence?

A content provider generated with Schematic will be used. Shared preferences will be used for settings.

Describe any corner cases in the UX.

If there is no data to be shown, show a card asking for data or stating there is no data.
For first use show home card titles for the user.

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

Schematic - for easier implementation of content providers.
Butter knife - for easier view binding and event listener addition.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Create project
- Setup libraries
  - Setup ButterKnife
  - Setup Schematic
- Setup Google Play Services
  - Setup Admob
  - Setup Analytics
- Create activities
  - MainActivity
  - DeadlinesActivity
  - AgendaActivity
  - AccountabilityChartActivity

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
    - Extend AppCompat for all themes
    - Implement parallax scrolling
- Build UI for DeadlinesActivity
- Build UI for AgendaActivity
- Build UI for AccountabilityChartActivity
- Build UI for ProductivityQuizActivity
- Enable RTL layout switching
- Fix layout from smallest to largest screens

Task 3: Build Content Provider

- Create database and content provider using Schematic
- Create tables
    - Deadlines
    - Agenda
    - Accountability Chart
- Define content provider
    - Deadlines
    - Agenda
    - Accountability Chart

Task 4: Implement MainActivity

- Implement layout selection
- Populate card views using content provider
- Link detail activity for each card
- Implement share app function
- Implement show/hide card titles

Task 5: Implement DeadlinesActivity

- Implement layout selection
- Populate card views with content provider
- Scroll to next deadline or scroll to most recent deadline
- Add colors to deadlines
    - Gray for past deadlines

- ○ Red for next immediate deadline
- ○ Black for future deadlines
- Populate main activity deadline
  - ○ Get next deadline
  - ○ Show time left before next deadline
  - ○ Show placeholder text if no deadlines
  - ○ Restart CursorLoader when deadline is reached
    - ■ Do not restart if no next deadline
- Implement data entry and validation
  - ○ Implement add deadlines
  - ○ Implement delete deadlines
    - ■ Add delete button to layout
    - ■ Select list items
    - ■ Show delete button if selected
    - ■ Implement delete function
  - ○ Implement edit deadlines
    - ■ Add edit button to layout
    - ■ Show edit button if selected
    - ■ Implement edit function

Task 6: Implement AgendaActivity

- Implement layout selection
- Populate card views with content provider
- Implement monthly/daily view
- Implement data entry and validation

Task 7: Implement AccountabilityChartActivity

- Implement layout selection
- Populate card views with content provider
- Implement monthly/daily view
- Implement data entry and validation

Task 8: Implement Timer Widget

- Create widget layout
- Update widget views with timer service

Task 9: Intercept broadcast events

- Intercept text and add to agenda

Task 13: Implement Accessibility

- Add content descriptions
- Implement D-pad navigation

Task 14: Implement Google Services

- Implement AdMob
- Implement Analytics

Task 15: Create signing configuration

- Create keystore with wizard
- Reference keystore in build file via relative path

Add as many tasks as you need to complete your app.

Submission Instructions
1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"