

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**”

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Screen 3](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you’ll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Implement Data Persistence](#)

[Task 4: Implement Google Play Services](#)

[Task 5: Implement Widget](#)

[Task 6: Implement Saved State with IcePick](#)

[Task 7: Test and Handle Error Cases](#)

GitHub Username: [chrisconnolly](#)

Web Browser for Kids

Description

Most web browser apps are dangerously open and can potentially allow your children to stumble upon questionable content. The Web Browser For Kids ensures that you, the parent, manage exactly what your child sees, when they can see it and for how long. The Web Browser for Kids app will allow children to browse the web safely due to its enhanced parental controls.

Intended User

My intended user is children aged 0-18.

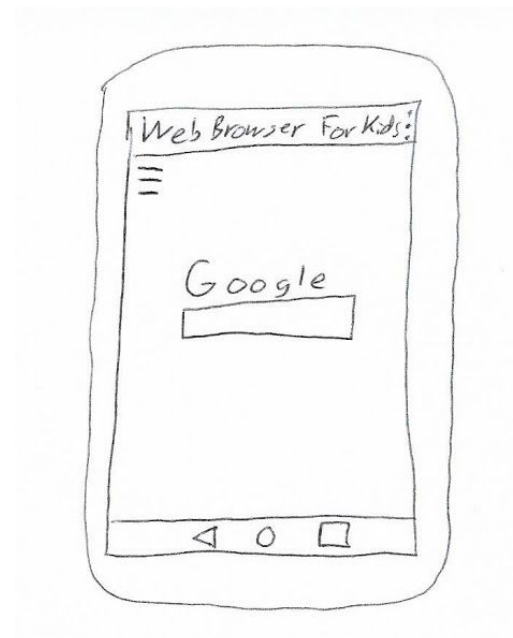
Features

The main features of the Web Browser for Kids includes:

- Website 'whitelist' - children can only visit websites entered/approved by a parent
- Curfew that will close the app after a configurable time
- Daily time limit for browsing to limit overuse
- 'Parent Mode' to allow parents to configure browser settings

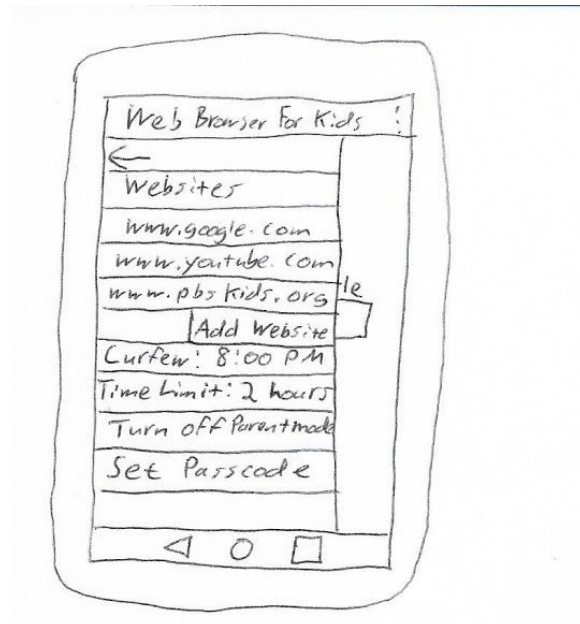
User Interface Mocks

Screen 1



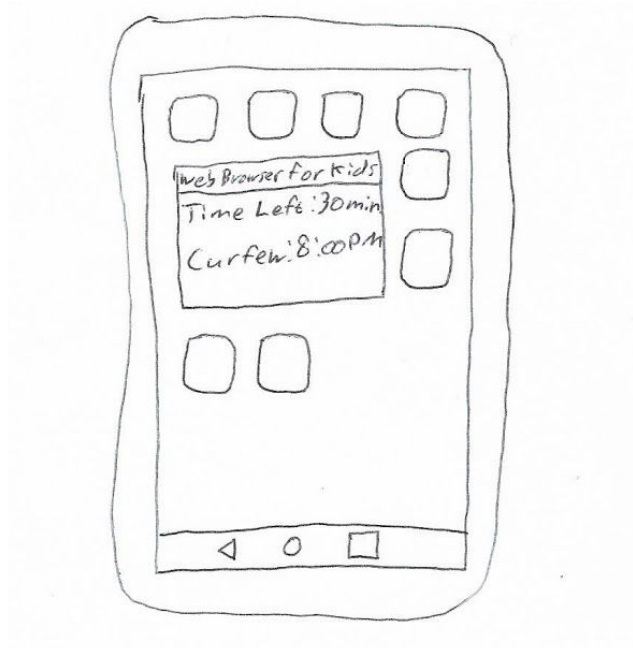
This is the main browser screen. The screen consists primarily of a WebView to show the online content. There is an AppBar at the top and two buttons ('sandwich' icon in top left and menu icon top right) which will open the configuration screen.

Screen 2



This is the configuration screen where kids can select the website they'd like to browse to. Parents will also use this screen to configure the app by adding websites, setting a curfew, time limit and passcode. Parents will be able to turn on 'Parent Mode' by entering the passcode they've set so they can configure all settings. No settings may be changed while not in 'Parent Mode'.

Screen 3



This is the widget for the home screen which will show how much time is left from the time limit. If there is 30 minutes left, the child may use the web browser for 30 more minutes total before the curfew for that day. After the curfew, no websites are accessible unless in 'Parent Mode'.

Key Considerations

How will your app handle data persistence?

Data persistence will be managed with `SharedPreferences` and a `ContentProvider`.

Describe any corner cases in the UX.

Corner Case 1: User cannot access a website they like.

1. The parent will login in 'Parent Mode' using a pass-code
2. Parent will add the website in question
3. Parent will exit 'Parent Mode'

Corner Case 2:

1. While browsing, user wishes to exit the application.
2. User presses the home button

Corner Case 3:

1. User wishes to access the last web page they visited
2. User touches the back button

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

I will be using IcePick to simplify saving my instance state (<https://github.com/frankiesardo/icepick>). This will eliminate boilerplate code in saving/restoring the instance state allowing my app to better respond in offline/airplane mode.

Describe how you will implement Google Play Services.

I will use AdMob to provide advertisements for the user. I will use Google Analytics to help implement the time limit feature.

Next Steps: Required Tasks

Task 1: Project Setup

- Create new Android application in Android Studio
- Import IcePick 3rd party library from GitHub
- Setup Google Services for AdMob and Analytics
- Build UI XML
- Build Data Persistence using SharedPreferences and ContentProvider
- Wire up UI with Data Persistence
- Implement Business Logic
- Build front-end widget

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for ConfigurationActivity
- Build UI for widget

Task 3: Implement Data Persistence

- Save all configurations to SharedPreferences
- Create ContentProvider to reference SharedPreferences

Task 4: Implement Google Play Services

- Import libraries for AdMob and Analytics
- Create UI for AdMob ads
- Implement code logic for Analytics

Task 5: Implement Widget

- Create XML UI for widget

- Create RemoteViewService to update widget
- Write code to wire widget comms with main application

Task 6: Implement Saved State with IcePick

- Import IcePick library and build with gradle
- Write code to save/restore instance state with IcePick

Task 7: Test and Handle Error Cases

- Test all Use Cases for applicable devices
- Consider error cases
- Ensure all error cases are handled using try/catch blocks and logging

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**"