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: krkeco

It's A Date

Description

It's a Date is designed for busy people who have busy schedules
Simply make a QR code, and anyone with the code can add into the event
Set up your own 'available' timeframes and, using Google Calendar, It's A Date will find the best times within your timeframes.

Intended User

Anyone who needs a hand with planning meetings, parties, etc.

Features

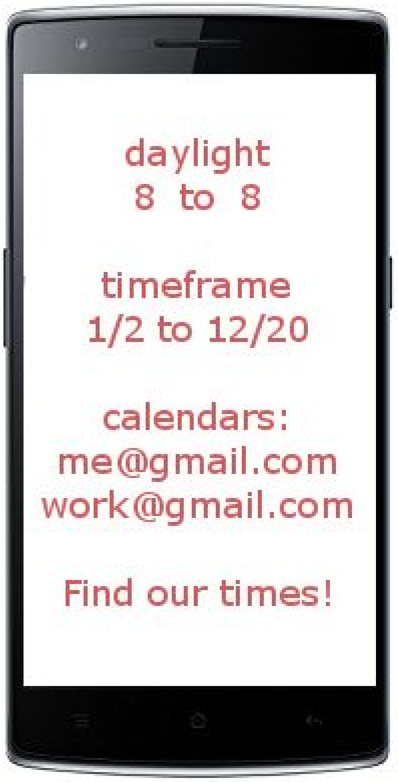
List the main features of your app. For example:

- Connect to local calendars via cronofy library
- Send your ‘availability’ to the firebase realtime database server
- Return times that are best for the group
- Available in app purchase of recurring widgets (to save group infos and simplify setup process)

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.

Screen 1



daylight
8 to 8

timeframe
1/2 to 12/20

calendars:
me@gmail.com
work@gmail.com

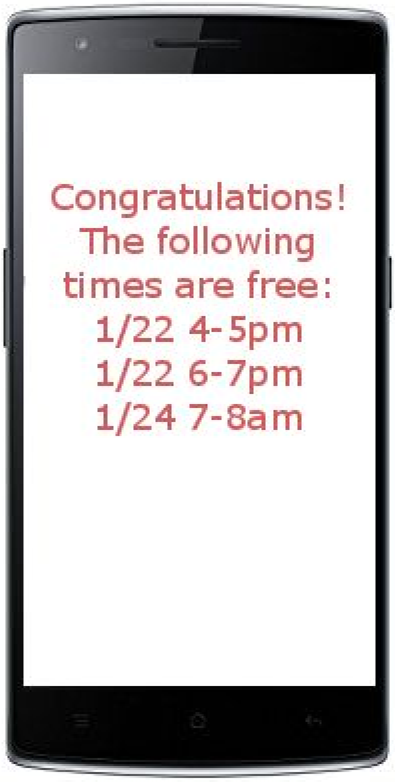
Find our times!

Daylight: is the time each day you are willing to meet from (8am to 8pm for example, this may need to be adjusted for work)
2 dropdown number pickers will provide the interface for this.

Timeframe is the days of time you are looking to meet within (maybe anytime from tomorrow to next Thursday)
2 month pickers will provide the interface for this

Calendars:
At the start of the activity, there will be a check for the calendar and it will retrieve all the calendars stored on the device, and then populate a listview or recyclerview of them with a checkbox.

Lastly the Find our time button will make an asynctask that will make a check of all calendars checkboxed, and retrieve the calendar information they have, make a freespace list and send that to the server



Congratulations!
The following
times are free:
1/22 4-5pm
1/22 6-7pm
1/24 7-8am

The server will return screen 2 with the free times available going in soonest first order

Widget will provide user with easy access to default settings from another phone or group of phones for repeat functions. It will retain the database_id for future use.

Key Considerations

How will your app handle data persistence?

There will not be persistent data because that might be creepy
The firebase database will keep blank databases with ids that are same so people can reconnect (it will clear data after event cleared)

Describe any corner cases in the UX.

When a group doesn't actually have any available time in their schedules

Ensure you can't go back in time on dates, but still have the remainder of 'today' available

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

Not sure I will need any

Describe how you will implement Google Play Services.

Will incorporate the Calendar API to retrieve calendar dates, and create them (in the event of a successful day being chosen).

Next Steps: Required Tasks

Task 1: Project Setup

- Create project
- Add calendar api
- Add firebase module
- Add in app purchases
- Add cronofy lib

Task 2: Implement UI for Each Activity and Fragment

- Create UI with textviews and button
- Create UI for returnfragment view of asynctask
- Create widget button
- Create notification for when data is sent to db from other phones

Task 3: Create calendar data classes

- Implement cronofy calendar api to retrieve data via loader
- Take events and combine into 'free time' slots
- Package data in a way that is easily stored in firebase
- method to create a calendar event under the 'default' calendar

Task 4: Create Async/Backend

- Send data to firebase via asynctask (send eventID, participants#, count, subarray[freetime])

- Create datareceivelistener for when the data gets processed and sent back (triggers second activity returns available dates in list)
- Enable data compile
  - Put subarray of free dates
  - Count +=1 and check participant# vs count, if same: download db and process in a asynctask
  - Return the entries that meet criterion to all participants via notification

---

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**"