

Description .....	1
Intended User.....	1
Features.....	2
User Interface Mocks.....	2
Screen 1.....	2
Screen 2.....	3
Key Considerations.....	3
How will your app handle data persistence? .....	3
Describe any edge or corner cases in the UX.....	3
Describe any libraries you'll be using and share your reasoning for including them.....	3
Describe how you will implement Google Play Services or other external services.....	3
Next Steps: Required Tasks.....	4
Task 1: Project Setup.....	4
Task 2: Implement UI for Each Activity.....	4
Task 3: Linking Activities to each other.....	4
Task 4: Buildout Other Activities.....	5
Task 5: RegisterActivity/LoginActivity/LogoutActivity buildout.....	5
Task 6: QuestionActivity/CorrectAnswerActivity/WrongAnswerActivity buildout.....	5
Task 7: AdMob implementation.....	6
Task 8: Database implementation.....	6
Task 9: Prepare app for release.....	6

**GitHub Username:** [dburgnerjr](#)

## GoodForce

### Description

GoodForce is a trivia game inspired by Givling with a focus of eliminating credit card debt, car, personal and payday loans. Using the power of crowdfunding through the purchase of GoodForce coins, extra lives and ad revenue with a goal of \$15,000 per funding effort.

### Intended User

This app has an intended audience for those who have credit card debt, car, personal and payday loans and would like to see it go away quickly.

### Features

The main features of your app include:

- Ability to ask user questions, getting strikes if they answer a question incorrectly and an extra life if get ten questions in a row correct. If a user gets three strikes and they have no extra lives, their round is over. If a user has extra lives, they will be allowed to use one extra life per round.
- Ability to accumulate GoodForce points based on: trivia scores, purchasing GoodForce coins, extra lives, watching an ad at the conclusion of each game or at an hourly interval.
- Leaderboard for scores and GoodForce points. The highest trivia score at the end of the trivia

period will win a prize while the user with the highest amount of GoodForce at the end of the funding effort will be the next person that will be the subject of the next funding effort.

## 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](http://www.ninjamock.com), Paper by 53, Photoshop or Balsamiq.

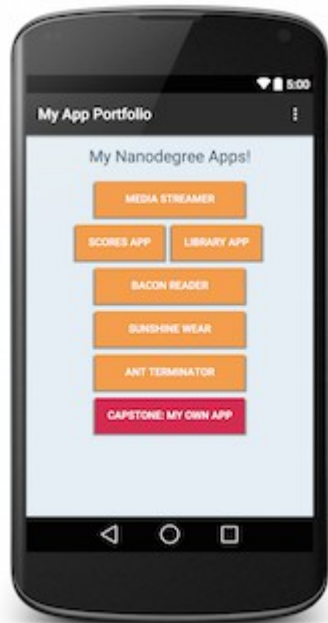
### Screen 1



Replace the above image with your own mock [ click on the above image, then navigate to Insert → Image... ]

Provide descriptive text for each screen

## Screen 2



Replace the above image with your own mock [ click on the above image, then navigate to Insert → Image... ]

Provide descriptive text for each screen

Add as many screens as you need to portray your app's UI flow.

## Key Considerations

**How will your app handle data persistence?**

The question/answer bank is be stored in a database that will be accessed via Google Cloud Environment. User information will be stored in a separate table in same database

**Describe any edge or corner cases in the UX.**

Once game play starts, the user will not be allowed to go back to previous question if he/she missed it. When ad video starts playing, upon clicking the video the user will go to the Play Store link.

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

Will be using Butterknife for data binding information handled during the course of the game.

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

This app will use AdMob for the purpose of playing videos to help the user score GoodForce points so they can move up in the leaderboard. In addition, will also be using Google Cloud Environment to facilitate data handling (see above).

## Next Steps: Required Tasks

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

### Task 1: Project Setup

The plan is to setup the project in Android Studio by configuring the build files to include Butterknife, AdMob and Google Cloud Environment.

Subtasks will include:

- Setting up build files to link with libraries as above
- Initial testing

### Task 2: Implement UI for Each Activity

These screens will be created:

- MainActivity – Intro interface
- GameStartActivity – Initial game interface to include stats (coins, extra lives, points)
- QuestionActivity – Once user hits the “Play” button, this interface displays the question, time remaining, point value, “True” or “False” buttons, score and number of strikes.
- CorrectAnswerActivity – Displayed when user answers question correctly.
- WrongAnswerActivity – Displayed when user answers question incorrectly.
- RegisterActivity – Displayed when a new user fills out info for the first time.
- LoginActivity – Displayed when a user logs in.
- LogoutActivity – Displayed when user logs out.
- GameRulesActivity – Displayed game rules.
- AboutActivity – What is this app and its mission in life.
- WatchVideoActivity – Used to watch videos at certain times (hourly intervals, when game is over).
- CoinsActivity – Where user can purchase coins (purchases disabled for purpose of this project).
- ExtraLivesActivity - Where user can purchase extra lives (purchases disabled for purpose of this project).
- PointsActivity – Displays how user has accumulated points.
- TellFriendActivity – Displays sharing interface because #SharingIsCaring

### Task 3: Linking Activities to each other

Linking each activity to each based on how this app is going to be setup.

These activities will be linked as follows:

- MainActivity: GameStartActivity, RegisterActivity, LoginActivity

- GameStartActivity: QuestionActivity, GameRulesActivity, AboutActivity, WatchVideoActivity, CoinsActivity, ExtraLivesActivity, PointsActivity, TellFriendActivity, LogoutActivity
- QuestionActivity: CorrectAnswerActivity, WrongAnswerActivity
- CorrectAnswerActivity: QuestionActivity (when game is not over), WatchVideoActivity (when game is over)
- WrongAnswerActivity: QuestionActivity (when game is not over), WatchVideoActivity (when game is over)
- All others: Will link back to GameStartActivity via “Back” button.

## Task 4: Buildout Other Activities

Buildout other activities from above that link back to GameStartActivity with Back button:

- GameRulesActivity/AboutActivity: Write a description of each activity and how it relates to the game.
- PointsActivity: Setup table that will display how the user is accumulating points.
- CoinsActivity/ExtraLivesActivity: Description of coins/extra lives and button menu to purchase either (actual purchases disabled since this will be a proof of concept app).
- WatchVideoActivity: Plays video and displays message to user upon completion of video.
- TellFriendActivity: Write a brief description with email address box inviting friend to download the app and start playing.
- Include “Back” button to go back to GameStartActivity.

## Task 5: RegisterActivity/LoginActivity/LogoutActivity buildout

Building out RegisterActivity/Login Activity/LogoutActivity:

- RegisterActivity: User registration form that will send email to the user's email address once user completes registration. Once completed, user goes back to the MainActivity.
- LoginActivity: User enters username and password (from RegisterActivity).
- LogoutActivity: User receives thank you for playing message and sends him/her back to the MainActivity.

## Task 6: QuestionActivity/CorrectAnswerActivity/WrongAnswerActivity buildout

Buildout of QuestionActivity/CorrectAnswerActivity/WrongAnswerActivity:

- QuestionActivity: Retrieve answer from user input within time allotted. If user does not answer within time allotted, it is automatically directed to WrongAnswerActivity. Questions will be selected from database at random.
- If user answers the question correctly, points will be awarded and the user gets the next question.
- Implement extra life reward check if user answers ten consecutive questions correctly, the user gets an extra life.
- Implement extra life use check if user answers three questions incorrectly, an extra life is consumed. If user has no extra lives and he/she has three strikes, the game is over.

## Task 7: AdMob/Google Cloud Environment implementation

Implement ads that will be used by this app:

- Register app with AdMob/Google Cloud Environment.
- Retrieve info and place it in the appropriate locations (WatchVideoActivity, QuestionActivity).

## Task 8: Database implementation

Implement database that will be used by this app:

- Build table that has user information.
- Build question/answer bank complete with explanation.
- Build table that shows how user has accumulated points.

## Task 9: Prepare app for release

Conduct final testing and proceed with Play Store listing:

- Create app icon.
- Add background images to make app presentable.
- Create Play Store listing.

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

## 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**”