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.

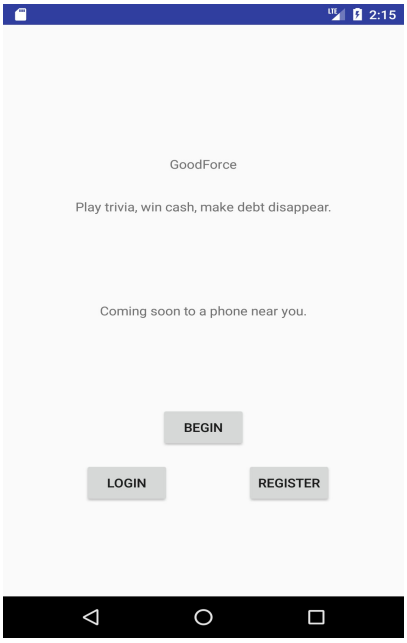
Features

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.

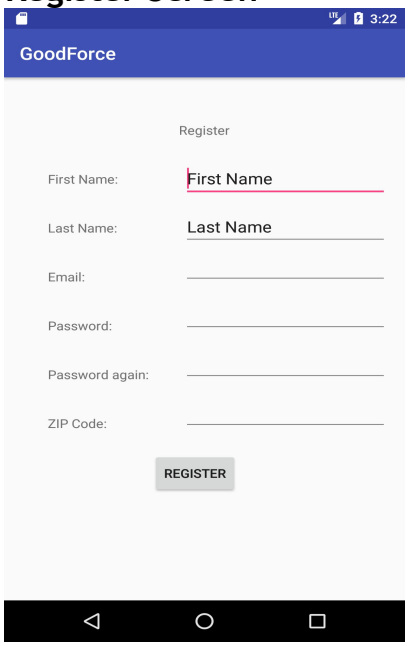
User Interface Mocks

Initial Screen



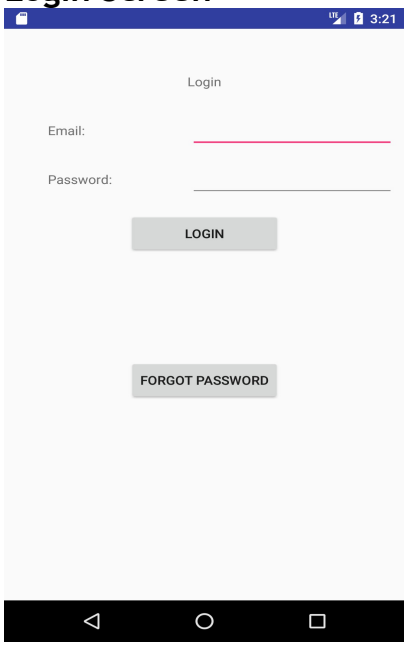
This is the first screen that the user sees.

Register Screen



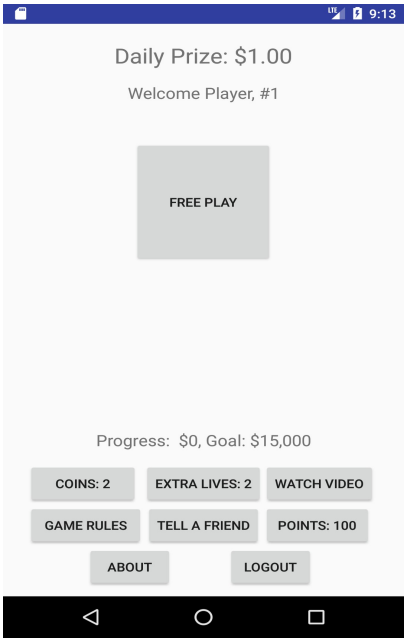
This is the register screen when the user signs up to become a GoodForce player.

Login Screen



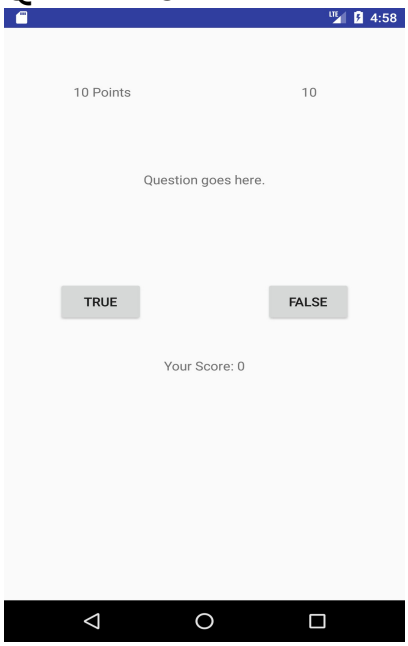
This is the login screen when the user logs in the play the game.

Game Screen



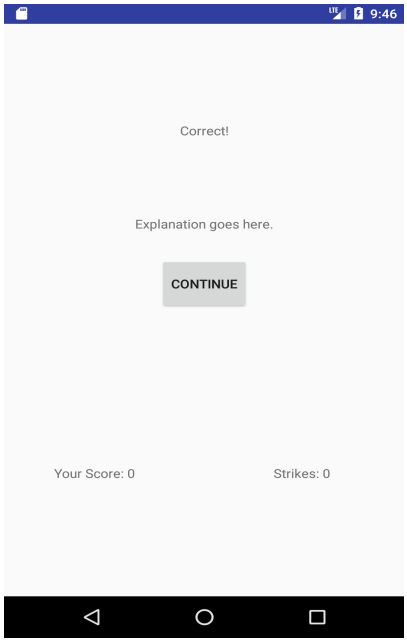
This is the game screen when the user has an opportunity to play.

Question Screen



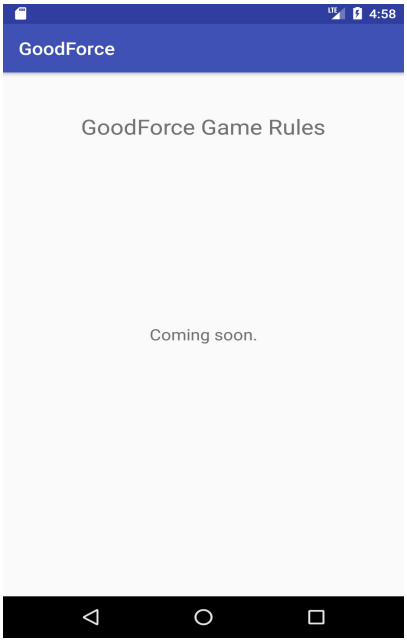
This is the where the user gets the question.

Answer Screen



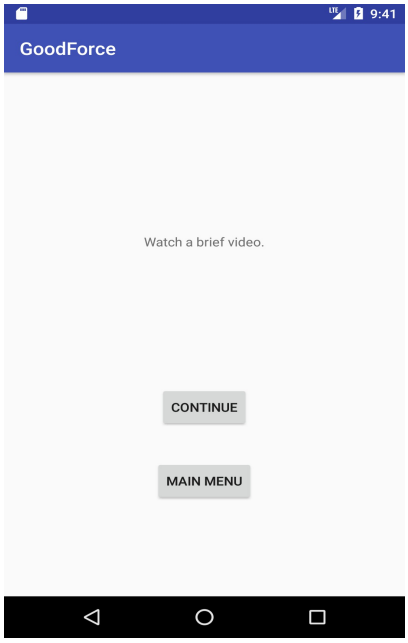
This is the where the user gets feedback and why.

Game Rules Screen



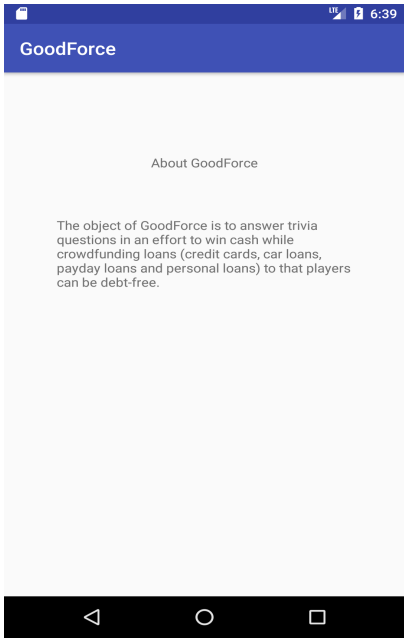
This is the where the user gets the game rules.

Watch Video Screen



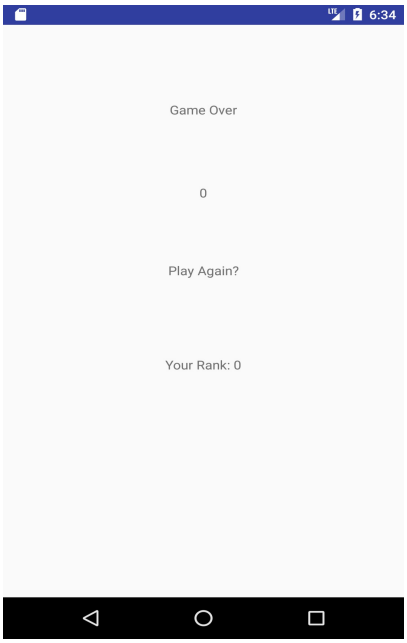
This happens when you watch video at the end of the game.

About Screen



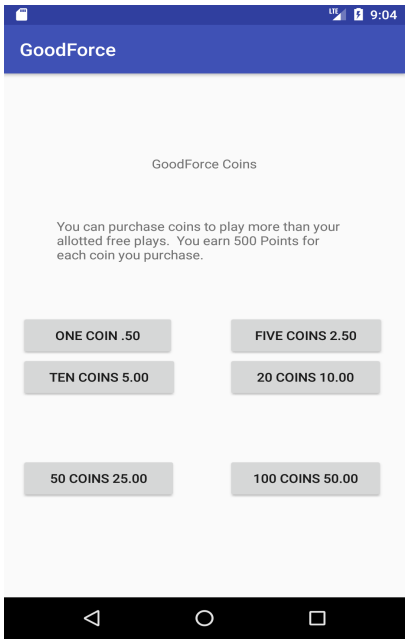
This is what this game is all about.

Final Score Screen



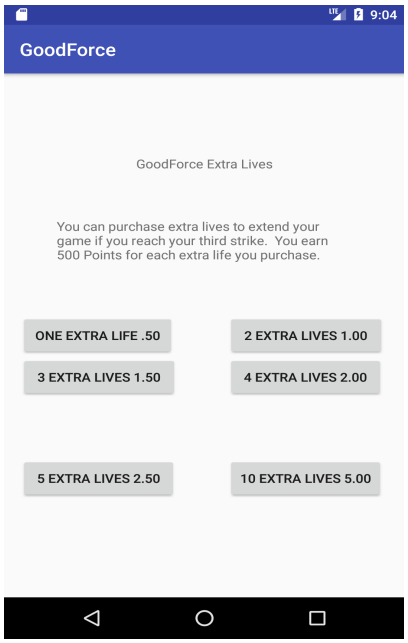
Displays the user's final score.

Purchase Coins Screen



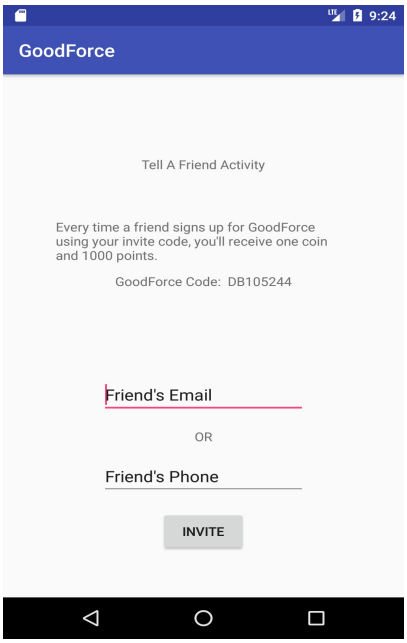
This is the where the user purchases coins.

Extra Lives Screen



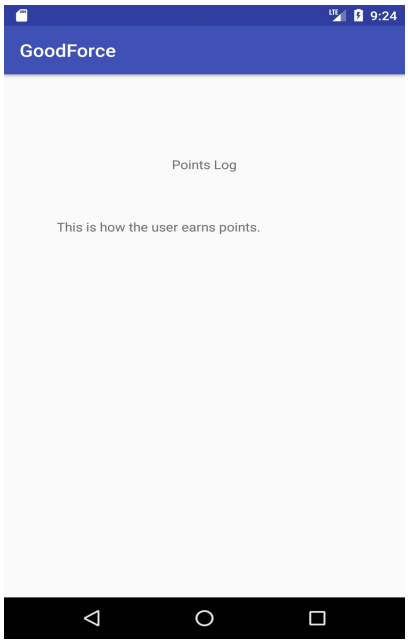
This is the where the user purchases extra lives.

Tell Friend Screen



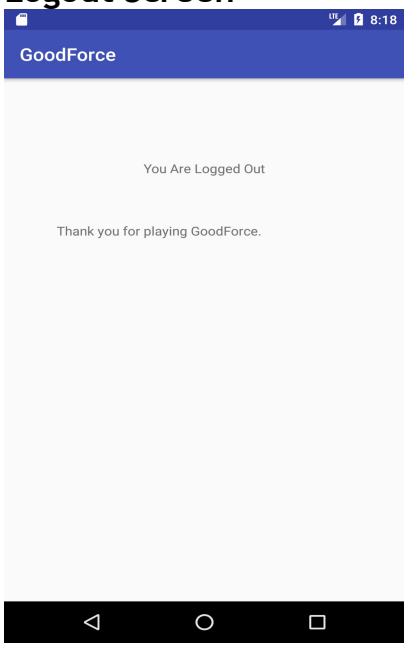
This is the where the user gets to share the app with others.

Points Screen



Shows how the user has earned points.

Logout Screen



Thanks the user for playing and logs the user out.

Widget

This is a widget view of this app.

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 using SQLite with Content Providers. AsyncTask will be used since the question bank will be accessed on an on-demand basis with Loaders to move data to the views. 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). Google Analytics will be used to track player usage.

Next Steps: Required Tasks

Task 1: Project Setup

The plan is to setup the project in Android Studio version 3.2 using Gradle version 4.4, Java as the development language by configuring the build files to include Butterknife version 8.8.1, AdMob (play-services-ads:15:0:1) and Google Analytics (play-services-analytics:10.2.4).

Subtasks will include:

- Setting up build files to link with libraries as above
- Initial testing
- App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.
- Accessbility will be addressed by using content descriptions.

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.
- AnswerActivity – Displayed when user answers question correctly/incorrectly.
- FinalScoreActivity – Displayed the user's final score at the conclusion of each game.
- 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: Implementation of App Widget

Buildout of App Widget in an effort to display a miniaturized version of app screen.

- Buildout AppWidgetProviderInfo which describes widget properties.
- Create BroadcastReceiver to build user interface.
- Add Widget configuration detailing how each activity will be converted into a widget.

Task 4: 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: AnswerActivity
- AnswerActivity: QuestionActivity (when game is not over), WatchVideoActivity (when game is over)
- WatchVideoActivity: FinalScoreActivity (at end of each game), GameStartActivity (otherwise)
- All others: Will link back to GameStartActivity via "Back" button.

Task 5: 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 6: RegisterActivity/LoginActivity/LogoutActivity buildout

Building out RegisterActivity/LoginActivity/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 7: QuestionActivity/AnswerActivity/FinalScoreActivity buildout

Buildout of QuestionActivity/AnswerActivity:

- QuestionActivity: Retrieve answer from user input within time allotted. If user does not answer within time allotted, it is automatically directed to AnswerActivity. Questions will be selected from database at random.
- Implement AsyncTask to access the question bank on an on-demand basis.
- 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.
- When the game is over, the user watches video and then gets their final score.

Task 8: 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 9: 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.
- Implement Content Providers using SQLite and Loaders to move data to views.

Task 10: 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.