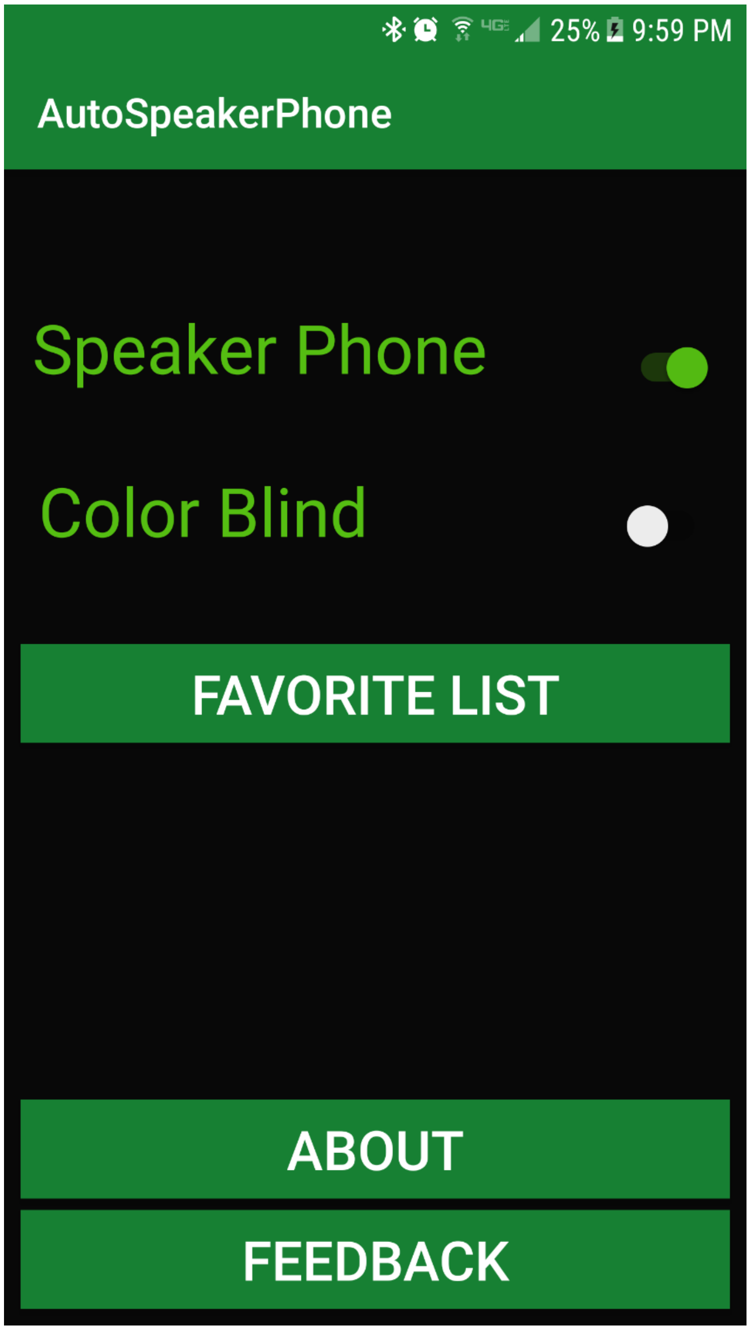


AutoSpeaker

THE FINAL STAGES

Henry Huynh, Sinna Uy, Gianluca Parilli, Brian Torok

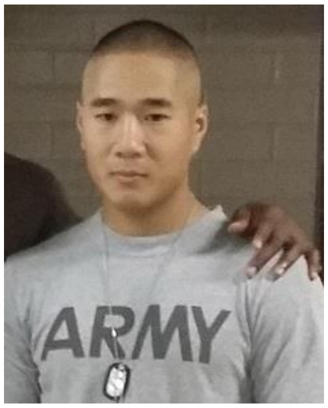
Client & Team Members

Here is a short bio of the client and all the members that worked on the project, and what roles they partook in.

Scott Warfield – Project Client/Mentor

Scott Warfield is a quadriplegic as a result of a diving accident. He is also a senior software developer for IBM with specialties in involving security and accessibility. After spending eight years in the military, he was hired by Internet security systems to be a developer on the X-Force RD team responsible for developing algorithms to detect malicious traffic. Scott currently spends a great deal of time working with people with disabilities in order to determine if there are software or hardware related solutions to assist them.

Sinna Uy – Lead Programmer and Tester



Henry Huynh – Database Modeler and Code Architect

Gialuca Parilli – UX/UI Designer

Brian Torok – Documentation Lead

Features and Bugs

Implemented:

- Turns on speaker phone when call is answered
- About page
- Database for feedback feature

To be implemented:

- Contacts list to automatically answer favorited contacts
- Colorblind Feature

Testing & Deployment

We have tested the app by way of unit testing, system testing, and user acceptance testing.

Unit Testing:

Unit testing involves testing of the methods getSpeaker() and speakerphoneSwitch() created with Junit.

System Testing:

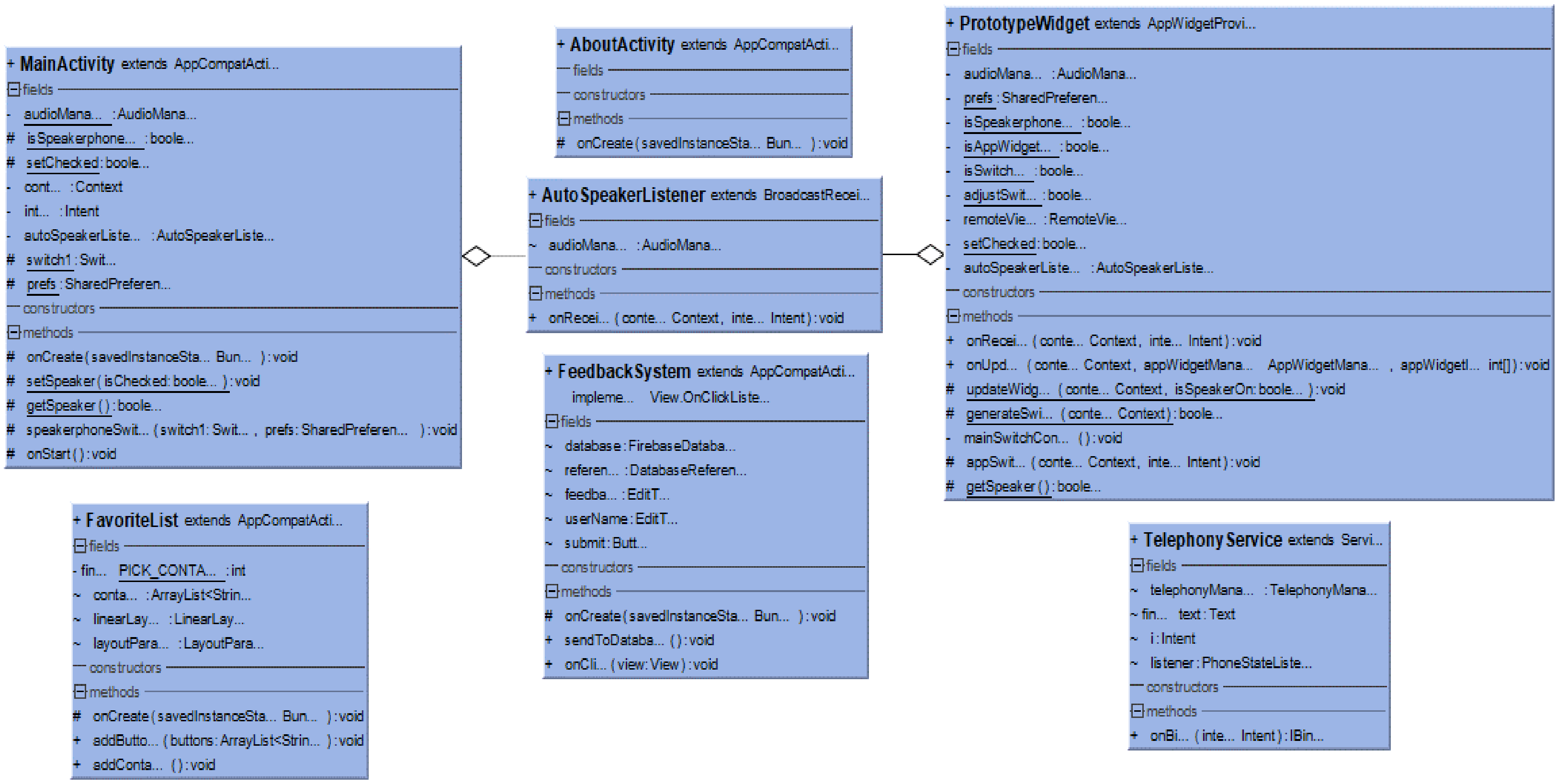
System testing involves testing on both emulated devices in Android Studio and testing on real devices in our possession. We tested mainly on Samsung devices ranging from android versions 5.0 to 7.0. A Galaxy S5, Galaxy S7, and the Galaxy Halo.

Deployment:

Our app is officially released into the Google Play Store. User information and changes to the application will all be located there.

Other Documentation & How to get the app

Here is a picture of the UML diagram for our java files.



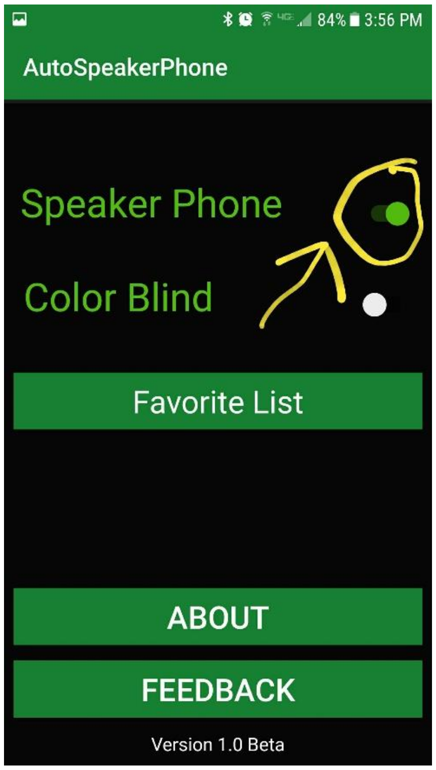
+ MainActivity extends AppCompatActi...
fields
- audioMana... : AudioMana...
# isSpeakerphone... : boole...
# setChecked : boole...
- cont... : Context
- int... : Intent
- autoSpeakerListe... : AutoSpeakerListe...
# switch1 : Swit...
# prefs : SharedPreferen...
constructors
methods
# onCreate (savedInstanceSta... Bun...) : void
# setSpeaker (isChecked : boole...) : void
# getSpeaker ( ) : boole...
# speakerphoneSwit... (switch1 : Swit... , prefs : SharedPreferen...) : void
# onStart ( ) : void
+ AboutActivity extends AppCompatActi...
fields
constructors
methods
# onCreate (savedInstanceSta... Bun...) : void
+ AutoSpeakerListener extends BroadcastRecei...
fields
~ audioMana... : AudioMana...
constructors
methods
+ onRecei... (conte... Context , inte... Intent) : void
+ FeedbackSystem extends AppCompatActi... impleme... View.OnClickListe...
fields
~ database : FirebaseDataba...
~ referen... : DatabaseReferen...
~ feedba... : EditT...
~ userName : EditT...
~ submit : Butt...
constructors
methods
# onCreate (savedInstanceSta... Bun...) : void
+ sendToDataba... ( ) : void
+ onCli... (view : View) : void
+ PrototypeWidget extends AppWidgetProvi...
fields
- audioMana... : AudioMana...
- prefs : SharedPreferen...
- isSpeakerphone... : boole...
- isAppWidget... : boole...
- isSwitch... : boole...
- adjustSwit... : boole...
- remoteVie... : RemoteVie...
- setChecked : boole...
- autoSpeakerListe... : AutoSpeakerListe...
constructors
methods
+ onRecei... (conte... Context , inte... Intent) : void
+ onUpd... (conte... Context , appWidgetMana... AppWidgetMana... , appWidgetI... int[]) : void
# updateWidg... (conte... Context , isSpeakerOn : boole...) : void
# generateSwi... (conte... Context) : boole...
- mainSwitchCon... ( ) : void
# appSwit... (conte... Context , inte... Intent) : void
# getSpeaker ( ) : boole...
+ FavoriteList extends AppCompatActi...
fields
- fin... PICK_CONTA... : int
~ conta... : ArrayList<Strin...
~ linearLay... : LinearLay...
~ layoutPara... : LayoutPara...
constructors
methods
# onCreate (savedInstanceSta... Bun...) : void
+ addButto... (buttons : ArrayList<Strin...) : void
+ addConta... ( ) : void
+ TelephonyService extends Servi...
fields
~ telephonyMana... : TelephonyMana...
~ fin... text : Text
~ i : Intent
~ listener : PhoneStateListe...
constructors
methods
+ onBi... (inte... Intent) : IBin...

Here is the poster that we will be using for the CREATE event.

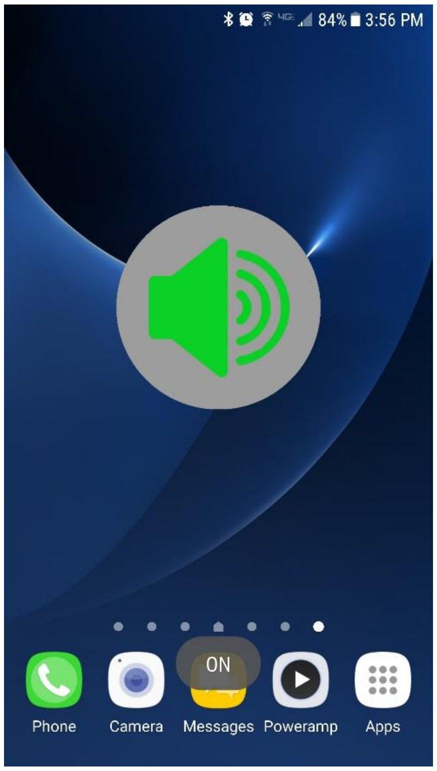
Auto Speaker Phone
DOWNLOAD NOW @ THE PLAY STORE
GET IT ON Google Play
Adobe Spark

The app is downloadable from the Google Play Store. All descriptions and user manuals will be located there.

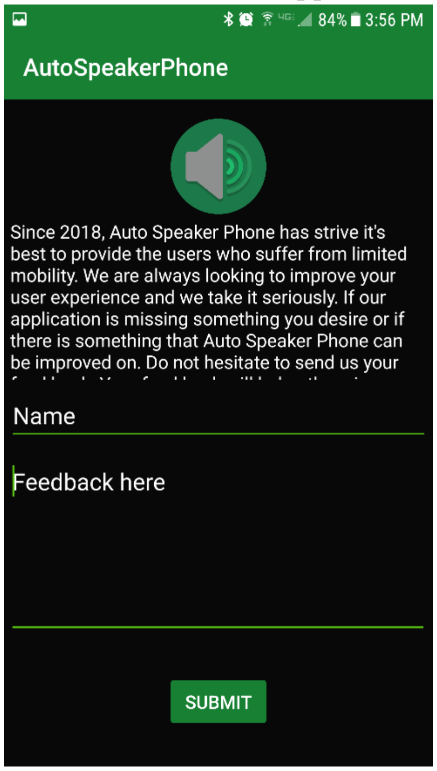
How to use the app



Tap the switch circled in the screenshot below to turn the app on and off.



You can also turn the app on and off via a widget on the home screen.



Since 2018, Auto Speaker Phone has strive it's best to provide the users who suffer from limited mobility. We are always looking to improve your user experience and we take it seriously. If our application is missing something you desire or if there is something that Auto Speaker Phone can be improved on. Do not hesitate to send us your

There is also a button on the main app screen to send feedback if the user has any issues.

Licensing

We have decided to go with a Creative Commons license. It will allow for free use of our app so long as it is not for commercial use and any derivatives must credit us. Attached will be the ownership agreement.

Performance Overview

As a preface to this section, we would like to admit fault to not properly utilizing the sprint tools we have been assigned, and therefore velocities and amount of time spent on some activities is either skewed or incomplete.

Sprints:

In no particular order:

1. create icons for the widget ON / OFF state
2. create telephony class to read the state of the phone calls
3. have working persistent data
4. favorite list
5. Research Android Studio's API
6. Test research in sandbox project
7. Push into GitHub once complete
8. Research Android Studio's API
9. Push into GitHub once complete
10. Widget on and off functionality
11. as a team member, i am going to upload the android studio to GitHub because that way we can all collaborate
12. As documentation lead, I need to create the presentation for the first iteration to properly inform those who need to know of what exactly our app will be capable of and how we plan to do it.
13. As a developer, I need to sync the main switch and appwidget together so both could toggle each other
14. Implement PhoneStateListener class
15. Disable Broadcast Listener from listening to phone's state when turned off
16. About Page
17. Implement Feedback System
18. Create Google Firebase database for feedback system
19. As developer, we need to sync the telephony service with the speaker state because we want the app to be on when the switch is on and vise-versa

Difficulty:

We used the Fibonacci sequence to give difficulty ratings to issues. On average, the difficulty was around a 5 or 8, with exceptions being given to 10, 13, and 19 which we would give a 13.

The overall project also turned out to be much more difficult than anticipated due to the confusing nature of how android devices manage their system usage and permissions.

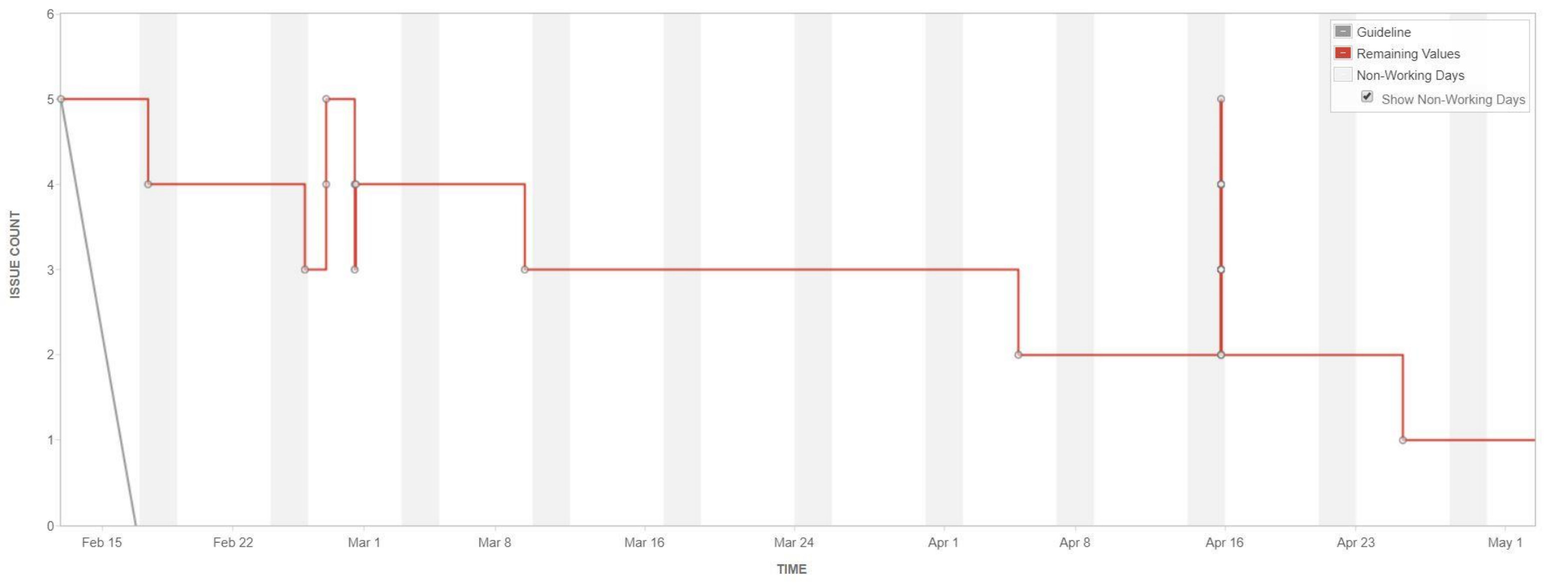
Individual Velocities

We did not properly document time spent on user stories and their difficulties, however the average amount of time spent on the easier stories labeled a 5 or 8 is about 2 weeks, with the more difficult issues such as 10, 13, and 19 taking about a month to complete.

Performance Reflection

Our biggest mistake in predicting how long a sprint would take is that we did not properly utilize Jira. Our second biggest mistake was underestimating this project. Had we properly utilized Jira, we might have been able to better predict how long some sprints would take, and properly allocate time to it. That would be our first step in better predicting sprint times. Our second step would be to simply get more experience. None of us had truly dealt with an android project like this, and given the work we had done in the past, we didn't expect this to be as difficult as it proved to be. In the future, better knowledge of the system would be very beneficial, especially if we could get the information from actual people who have worked with these tools previously.

Burndown Chart


Guideline
Remaining Values
Non-Working Days
Show Non-Working Days
ISSUE COUNT
TIME
Feb 15
Feb 22
Mar 1
Mar 8
Mar 16
Mar 24
Apr 1
Apr 8
Apr 16
Apr 23
May 1