Axiona Final Release Report

Team members

Name (Student id)	GitHub id	Story points
Adrianna Diaz (27184778)	**adriannadiaz**	**22**
Liuai Hatter (25976618)	ldhatter	17.5
Anna Rogozin (27494939)	annarog	16
Wahab Wajahat (21311980)	wahabwajahat	9.5
~~Francis Bouchard (26786812)~~	~~francisbouchard~~	~~0~~
~~James Talarico (40008054)~~	~~jtalarico~~	~~0~~
		65

Project summary

A tool to facilitate the management of notes that frontline social workers take when meeting participants (the people who they help). The software runs as a secure web application. The web application is able to sync and fetch relevant case files for the participants. To assist the frontline workers in managing tasks and organizing files.

Velocity

Iteration 9 (4 stories, 14 points)
Iteration 10 (4 stories, 11 points)
Iteration 11 (5 stories, 15 points)
Iteration 12 (4 stories, 16 points)
Iteration 13/Release 3 (3 stories, 9 points)

Velocity after Iteration 13 (Release 3 - Final): (14+11+15+16+9)/5 = 13
We completed three user stories for a total of 9 SP during this iteration.

URL: https://github.com/adriannadiaz/Axiona

Previous Project: DeepEye

We previously had a change of stakeholders, so the following is a summary of the implemented work for DeepEye.

Project Total: 14 stories, 101 points over 12 weeks

Iteration 1 (1 story, 8 points)
Set up frameworks and environment. Created a desktop application prototype to set directory of images to be used in the image classifier.

Iteration 2 (3 stories, 23 points)
Implemented the user interface so user can adjust parameters and labels of images to classify.

Iteration 3, (3 stories, 26 points)
Implemented an image classifier prototype using TensorFlow.

Release 1 aka Iteration 4, (4 stories, 23 points)
Integrated the image classifier prototype with the user interface. Application allows user to select images to train on, create and store models, and display classification prediction on an image.
Release 1 Total: 11 stories, 80 points over 10 weeks

Iteration 5, (3 stories, 21 points)
Application now returns predictions on multiple images. It was during this iteration that we left the stakeholder, Teledyne.

Current Project: Axiona

After changing stakeholders, the following describes work completed for Axiona.
Note: once changing projects, the scale of our user story points changed too.

Project Total: 30 stories, 111 points over 17 weeks

Iteration 6, (1 story, 13 points)
During this iteration, we began work with the new stakeholders, ASTTeQ. We completed initial surveys regarding what their current needs are. Based off their most needed requirement, we began implementing a social worker website. This iteration we completed: setting up the Angular client, Express server, Mongo database and participant creation.

Iteration 7, (5 stories, 15 points)
For this iteration we worked on features that allow a social worker to help a participant find a housing solution. This includes adding housing resources, managing a participant's profile, and assigning a participant to housing.

Release 2, Iteration 8, (4 stories, 18 points)
In this iteration we completed authentication for the users, adding notes related to participants, keeping track of participant logs, and creating new casefiles for participants.
Release 2 Total: 10 stories, 46 points over 7 weeks

Iteration 9, (4 stories, 14 points)
In this iteration we completed the addition of documents, the completion of the base case (assign participant to housing), and the creation of elevated level of user privilege.

Iteration 10, (4 stories, 11 points)
In this iteration we completed the validation for forms, the managing of user accounts, updating a casefile, and logging phone calls.

Iteration 11, (5 stories, 15 points)
In this iteration we completed the addition of different types of resources, the managing of deleted records, automatic username generation and the viewing and updating of the phonelog.

Iteration 12, (4 stories, 16 points)
In this iteration we completed the viewing and downloading of documents, uniformized the UI all throughout, with better navigation for the user and the application.

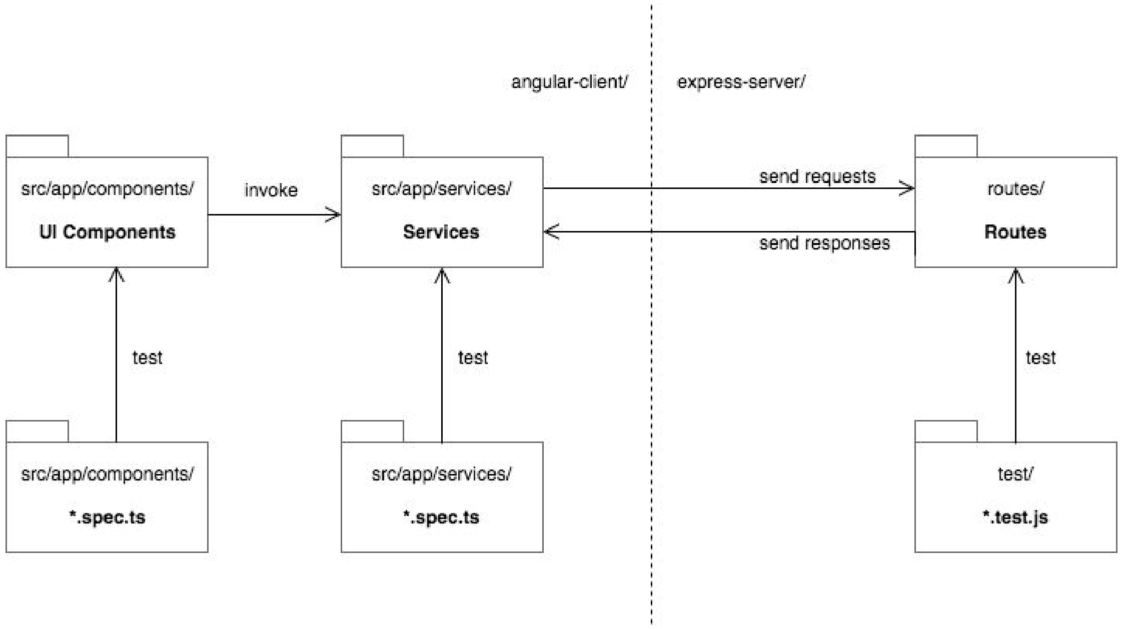
Iteration 13, (3 stories, 9 points)
In this iteration we completed the management of users, and the creation of reports, as well as fixing remaining bugs.
Final Release Total: 20 stories, 65 points over 10 weeks

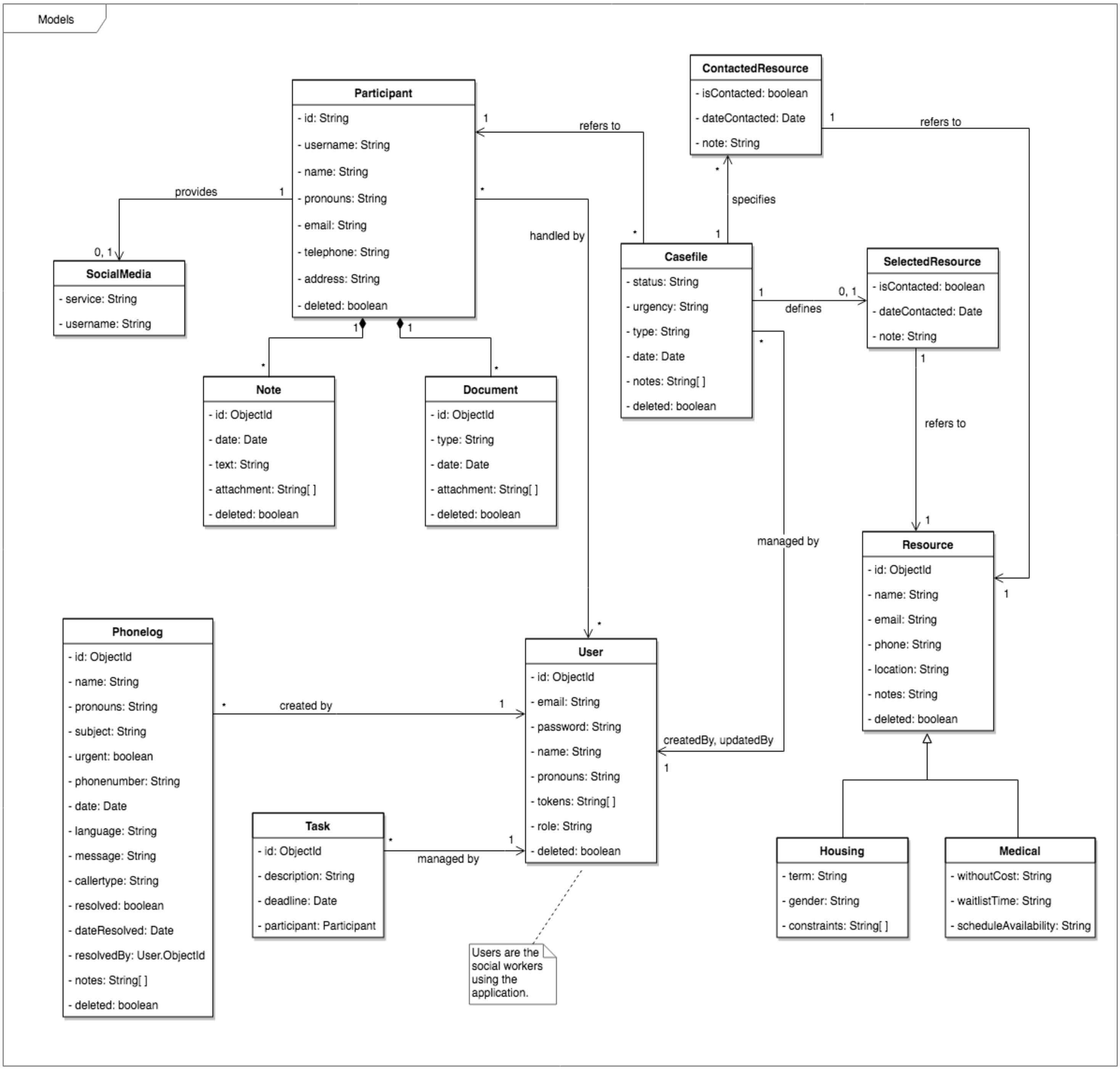
Main achievements of Final Release

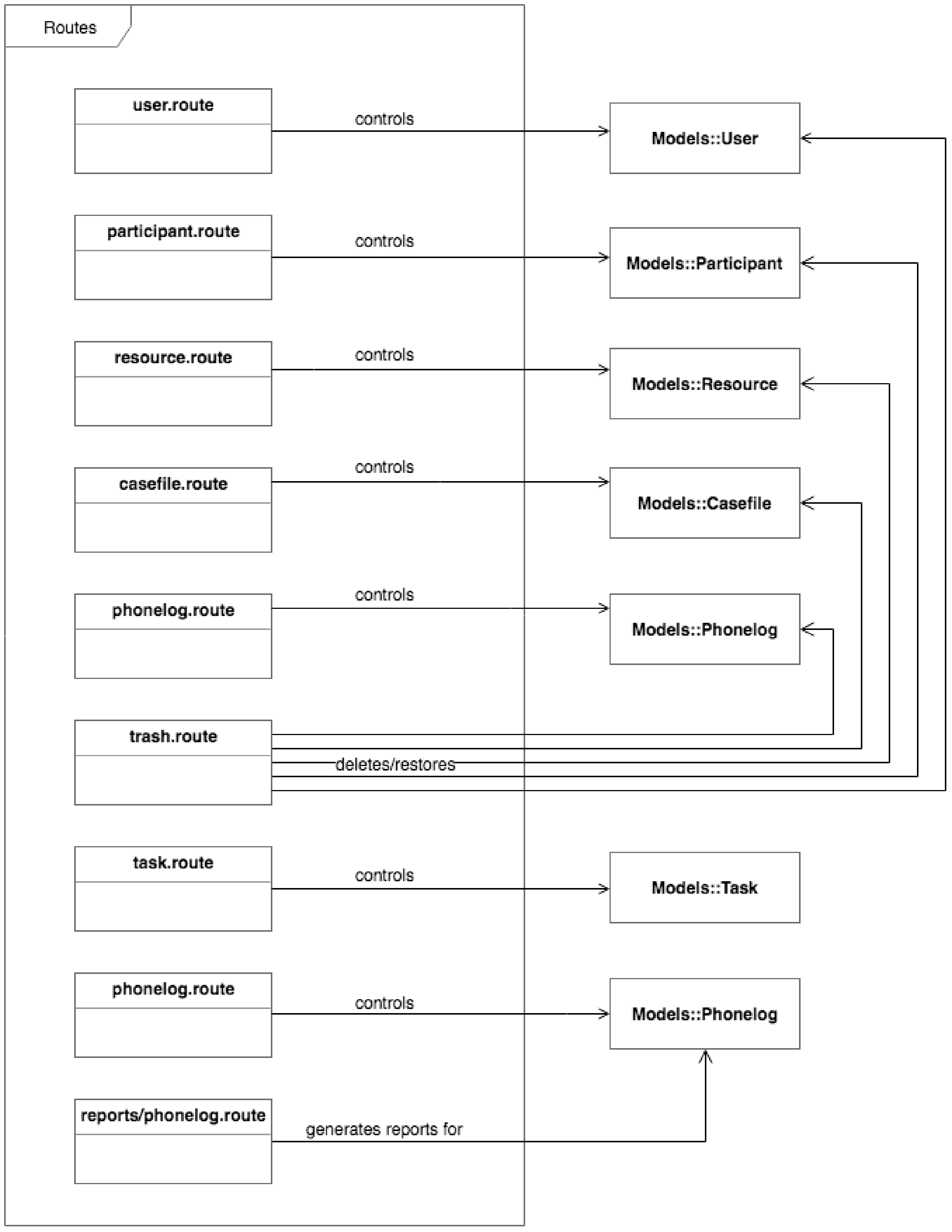
We have written an application that helps caseworkers handle their casefiles. While there is certainly the possibility of adding features to its modular architecture, the core functionality has been implemented.

Overall Arch and Class diagram

Here are the layers and main classes of our system:




Models
Participant
- id: String
- username: String
- name: String
- pronouns: String
- email: String
- telephone: String
- address: String
- deleted: boolean
ContactedResource
- isContacted: boolean
- dateContacted: Date
- note: String
refers to
1
*
refers to
1
provides
1
0, 1
SocialMedia
- service: String
- username: String
*
handled by
specifies
*
1
Casefile
- status: String
- urgency: String
- type: String
- date: Date
- notes: String[ ]
- deleted: boolean
SelectedResource
- isContacted: boolean
- dateContacted: Date
- note: String
1
defines
0, 1
*
1
1
*
*
Note
- id: ObjectId
- date: Date
- text: String
- attachment: String[ ]
- deleted: boolean
Document
- id: ObjectId
- type: String
- date: Date
- attachment: String[ ]
- deleted: boolean
1
refers to
1
managed by
Resource
- id: ObjectId
- name: String
- email: String
- phone: String
- location: String
- notes: String
- deleted: boolean
1
Phonelog
- id: ObjectId
- name: String
- pronouns: String
- subject: String
- urgent: boolean
- phonenumber: String
- date: Date
- language: String
- message: String
- callertype: String
- resolved: boolean
- dateResolved: Date
- resolvedBy: User.ObjectId
- notes: String[ ]
- deleted: boolean
*
User
- id: ObjectId
- email: String
- password: String
- name: String
- pronouns: String
- tokens: String[ ]
- role: String
- deleted: boolean
*
created by
1
createdBy, updatedBy
1
Task
- id: ObjectId
- description: String
- deadline: Date
- participant: Participant
*
managed by
1
Housing
- term: String
- gender: String
- constraints: String[ ]
Medical
- withoutCost: String
- waitlistTime: String
- scheduleAvailability: String
Users are the social workers using the application.


Routes
user.route
controls
Models::User
participant.route
controls
Models::Participant
resource.route
controls
Models::Resource
casefile.route
controls
Models::Casefile
phonelog.route
controls
Models::Phonelog
trash.route
deletes/restores
task.route
controls
Models::Task
phonelog.route
controls
Models::Phonelog
reports/phonelog.route
generates reports for

Infrastructure

List of Frameworks:

- Angular5 https://angular.io/
- Node https://nodejs.org/en/
- Express https://expressjs.com
- MondoDB https://www.mongodb.com

Infrastructural Changes:

After switching to a new stakeholder, our application domain changed. We no longer use TensorFlow, Electron, with the current application. Therefore, the

Name Conventions

List your naming conventions or just provide a link to the standard ones used online.

Database: We will follow the MongoDB Style guide
Backend: We will follow the Microsoft TypeScript Style Guide
Frontend: We will follow the Angular Style Guide.
Frontend styling: We will follow the Sass Guidelines.

Code

Key files: top **5** most important files (full path). We will also be randomly checking the code quality of files. Please let us know if there are parts of the system that are stubs or are a prototype so we grade these accordingly.

File path with clickable GitHub link	Purpose (1 line description)
angular-client/src/app/components/participants/participant-profile/participant-profile.component.ts	Handles participant management on the angular client.
angular-client/src/app/services/authentication.service.ts	Handles authentication (login, register heartbeat) in the angular client.
angular-client/src/app/components/casefiles/case-modal/case-modal.component.ts	Handles the creation of a new case for a Participant
express-server/routes/reports/phonelog.route.js	Handles the aggregation of data from the database to generate reports.
angular-client/src/app/components/reports/reports.component.ts	Handles the viewing of reports in the frontend.

Testing and Continuous Integration

Each story needs testing before it is complete. If some class/methods are missing unit tests, please describe why and how you are checking their quality. Please describe any unusually aspects of your testing approach.

List the **5** most important test with links below.

Test File path with clickable GitHub link	What is it testing (1 line description)
casefile.test.js	Testing that Casefiles are created and uploaded correctly in the database
participant.test.js	Testing that Participants are created and uploaded correctly in the database
resource.test.js	Testing that Resources are created and uploaded correctly in the database
user.test.js	Testing that Users are created and uploaded correctly in the database
participant-profile.component.spec.ts	Frontend testing for the component

We are using TravisCI, and the link is https://travis-ci.org/adriannadiaz/Axiona/builds/