



Thoughts

*Personal sentiment tracking &
analysis*

Software requirements

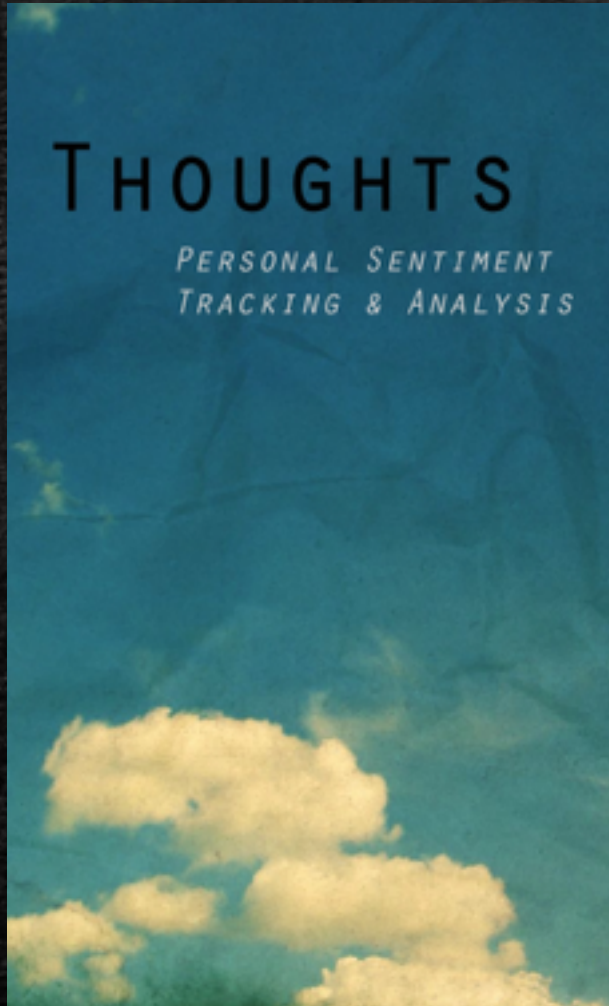
Yudhishtir Singh
Tarif Haque
Vlad Caciuc

Software Requirements Overview



- Problem definition & solution
- Scope of system & application architecture
- Overview of sentiment analysis requirements
- Features
- UML Diagrams
 - Use Case Diagram
 - Activity Diagram
 - High Level Class Diagram
- Project risks & challenges

Problem definition

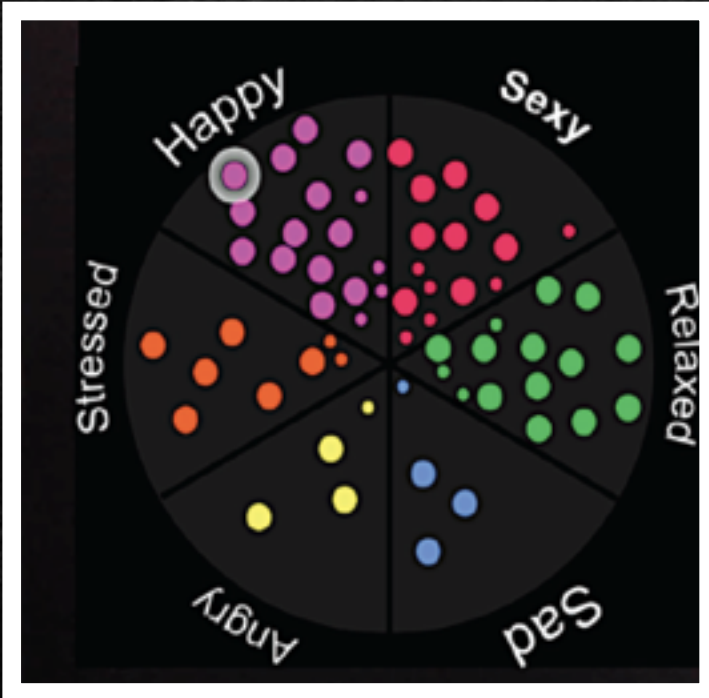


- How can a user track his or her **affective/emotional state**?
- How can an application achieve this task without directly asking the user about their emotional state?

Our solution

- The primary function of our application is to use **sentiment analysis** to classify and analyze user **thoughts** for positivity/negativity and affective/emotional state.

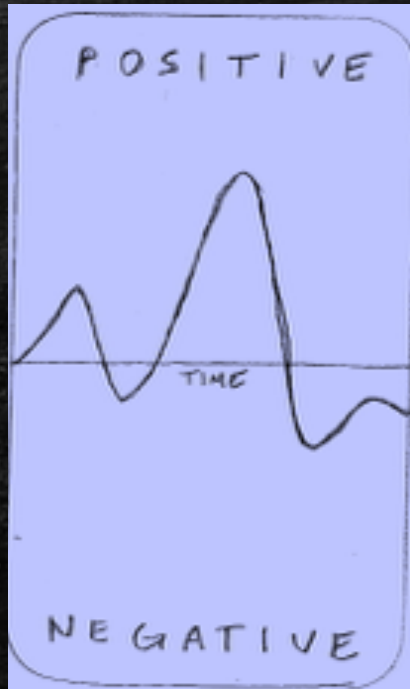
Scope of system



- The application will collect user *thoughts* throughout time
 - Mini “tweet-like” blurbs
 - Blog posts
 - Expressions of emotion like “I don’t feel well” or “Life is good”
- These thoughts will be *analyzed* and *classified*
- The application will present *data visualizations* of the resulting analyses to the user.

Analyses of thoughts

- **Polarity** (positivity/negativity) of thoughts, expressed as percentage, over time



- **Discrete emotion classification** – based on theory that all humans are thought to have an innate set of basic emotions that are universal and cross-culturally recognizable.

— Paul Ekman and colleagues (1972) concluded the six basic emotions are **anger**, **disgust**, **fear**, **happiness**, **sadness**, and **surprise**.



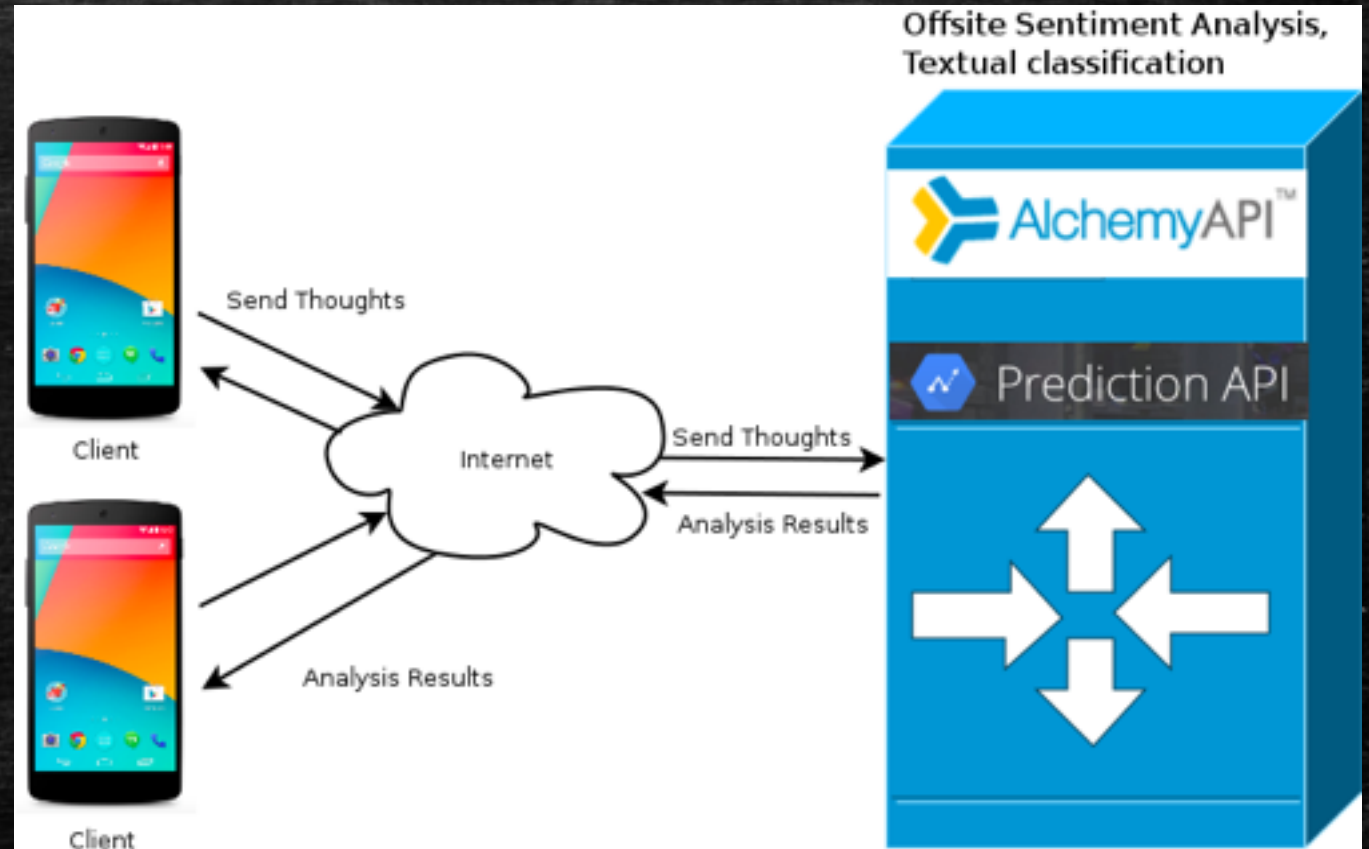
Creating a custom sentiment analysis model

1. Collect & label training data
 - "sad", "Feeling kind of low...."
 - "excited", "OMG! Just had a fabulous day!"
 - "bored", "Eating eggplant. Why bother?"
2. Upload training data to cloud
3. Train the model using Google Prediction API
4. Use trained model to classify unseen thoughts!



Application Architecture

- Client-server model
- RESTful API
- Mobile client sends requests to server for analysis/classification



Features

- Essential

- Login & registration screens
- Mechanism to add new thoughts / selecting thought(s) for analysis
- Mechanism to select different sentiment analysis models
- (1) Polarity visualization & (2) Discrete emotion classification visualization

- Possible

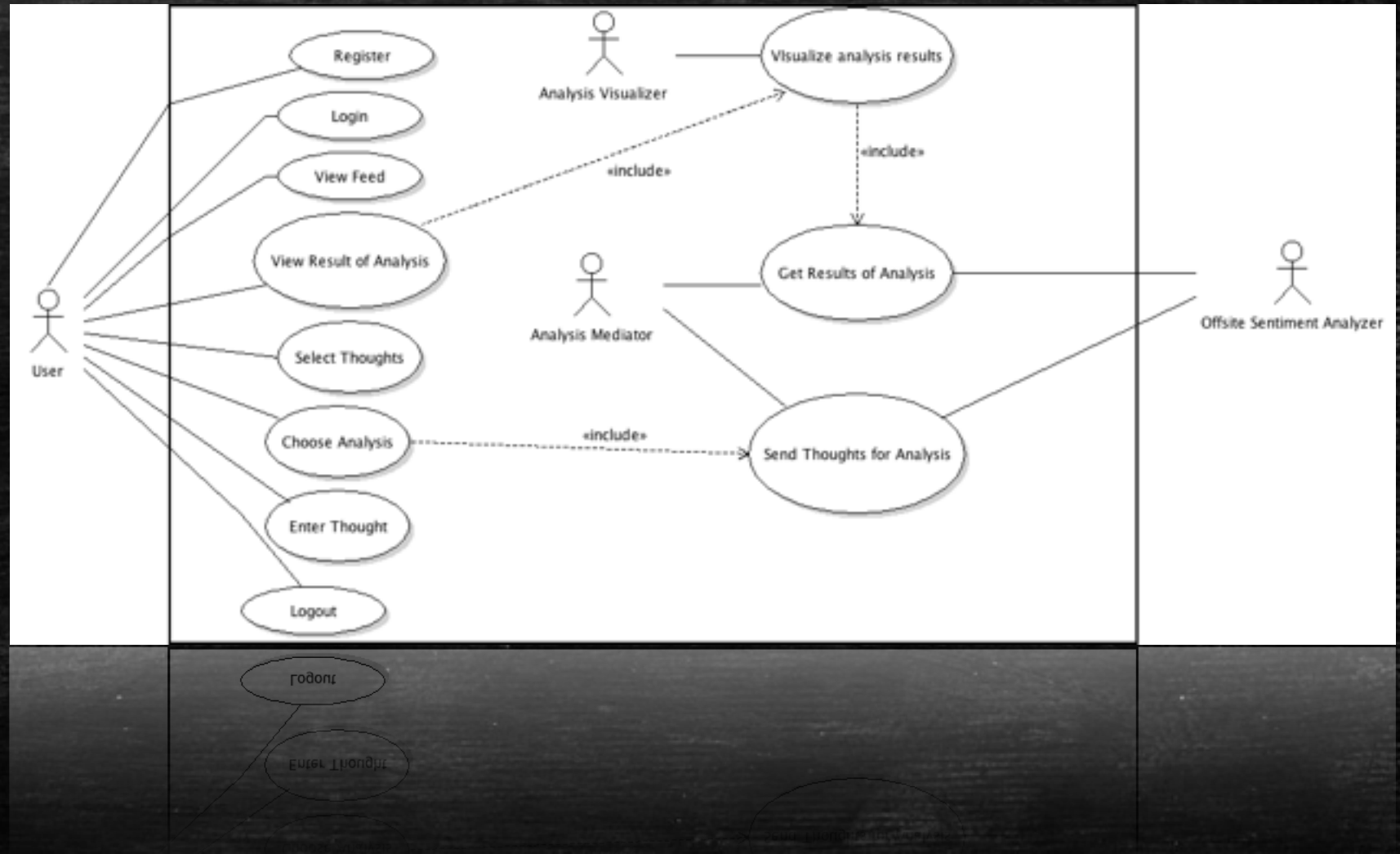
- “Mood map” visualization / other novel visualizations of thoughts
- Login into application and pull profile information from Google+

- If time available

- Develop more emotional classification models!

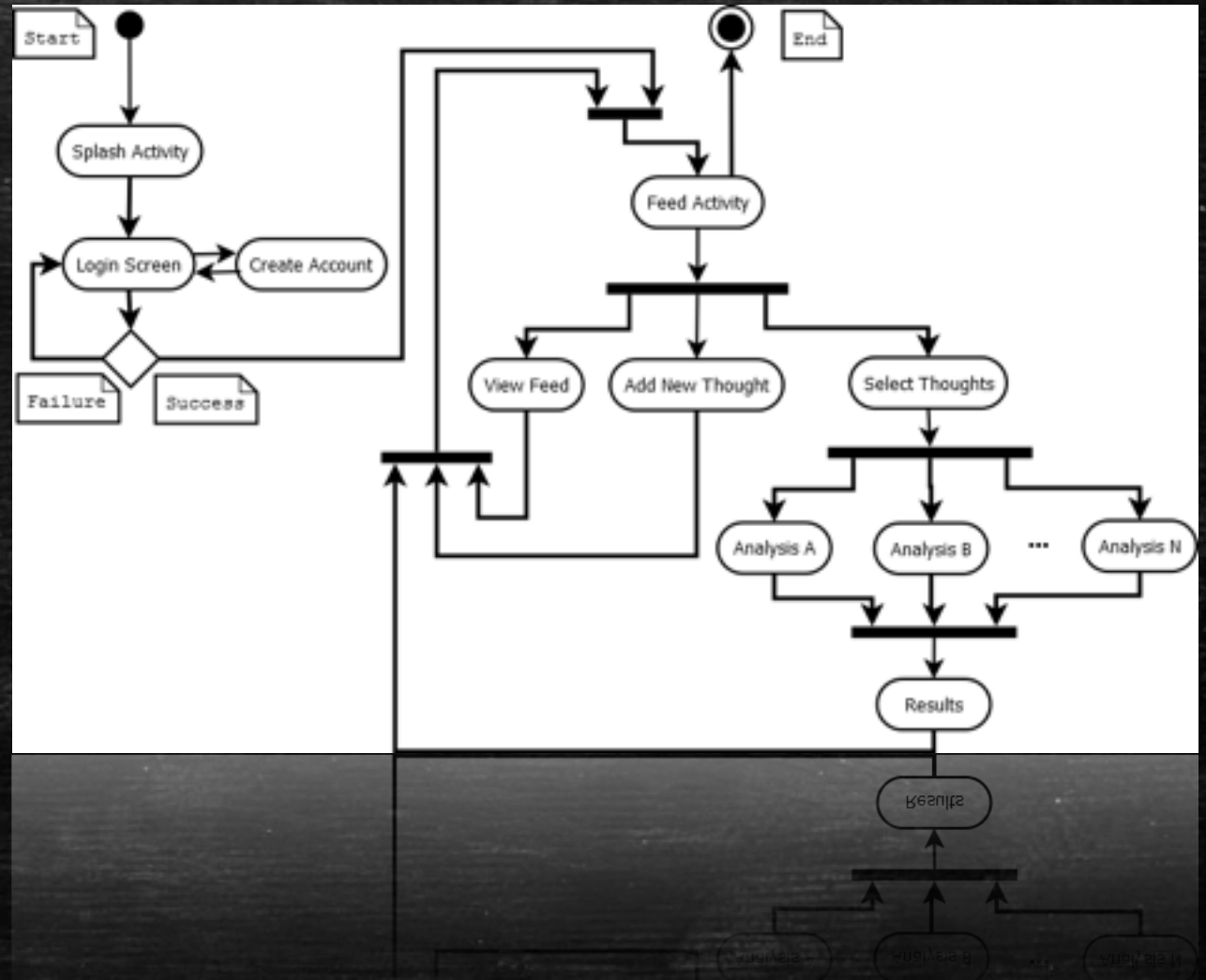
Use case diagram

- Three actors
 - User
 - Analysis Visualizer
 - Analysis Mediator
- Analysis Mediator sends *thoughts* to sentiment analyzer when user chooses an analysis
 - responsible for all communications with the server
- Analysis Visualizer draws graphs based on the results of analysis

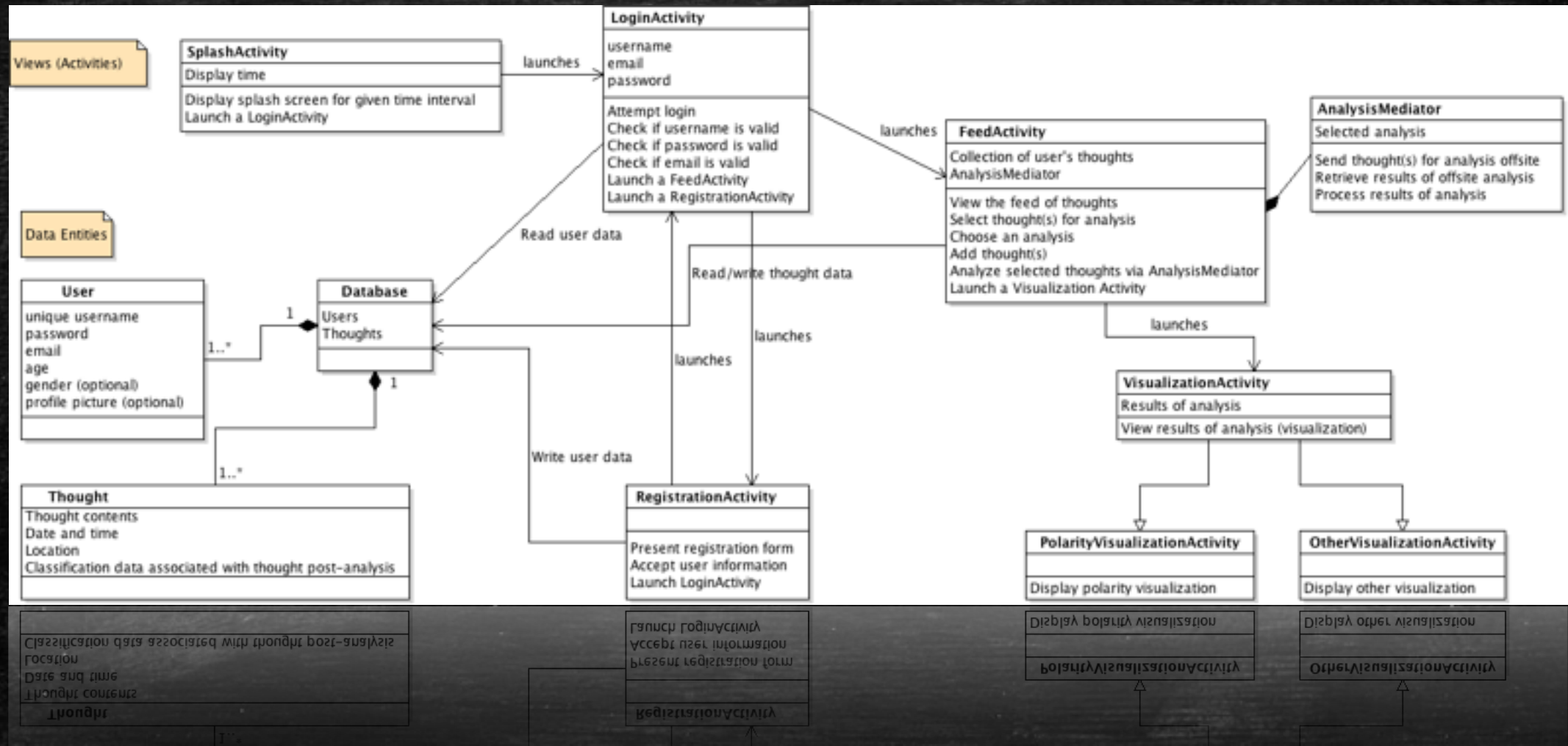


Activity diagram

- Login/register
 - Add new thoughts
 - Select thoughts for analysis
- Choose an analysis (e.g. polarity, discrete emotion classification)
- Visualize results of analysis



High level class diagram



Project Risks & Challenges

- People express **opinions** and **emotions** in complex ways
 - Collecting representative training data
- Primary hurdle: implementing sentiment analysis systems for different emotional classification models
 - Creating text classifiers based on present research in emotion classification may prove challenging
- Networking Android app with offsite sentiment analysis systems
- Google+ Integration

Questions?

