MICHIGAN STATE UNIVERSITY

Project Plan
Digital Banking with Chatbots

The Capstone Experience

Team MSUFCU

Syed Naqvi
Cori Tymoszek
Josh Benner
Chuanyun Xiao
Gustavo Fernandes

Department of Computer Science and Engineering
Michigan State University

Fall 2017

From Students…
…to Professionals

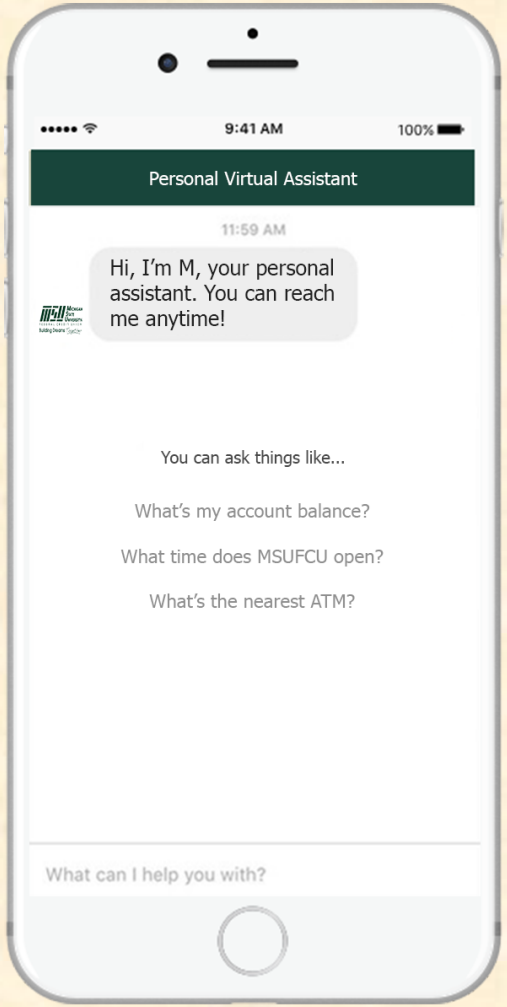
Functional Specifications

- Problem: Being Cost Effective while improving and maintaining a high level of customer service
- Solution: Allow users to perform basic tasks through Web, FB MSG, Google Assistant, and iOS
- How: Digital Chat Bot Assistant
- Hand off to live chat for complex tasks

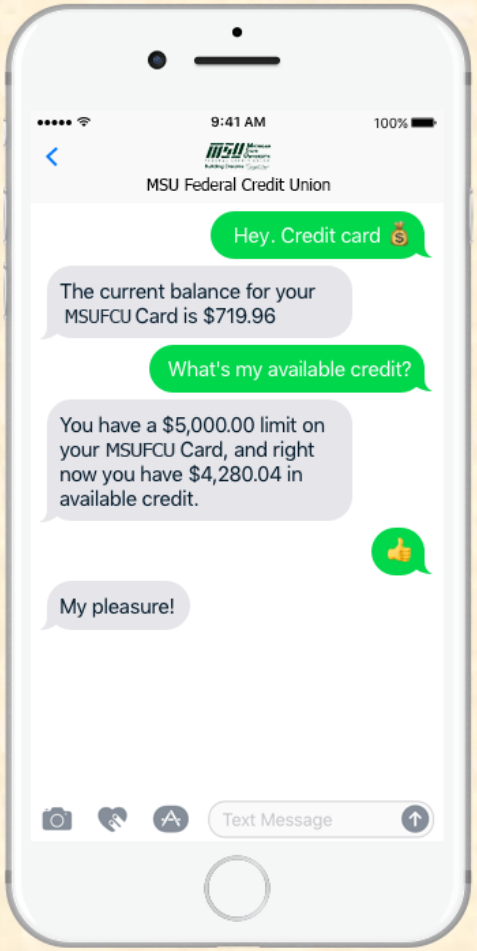
Design Specifications

- Handle:
  - Checking Balances
  - Transfers
  - Lost/Stolen Card
  - New loan/account questions, etc.
- Device interface(Web, FB Messenger, SMS, iOS)
- Voice Controls(Google Assistant, Alexa)

Screen Mockup: iOS and Android



Screen Mockup: iMessage and FB Messenger



Screen Mockup: Alexa

"Alexa, what's my account balance?"

"Alexa, I can't access my bank account."

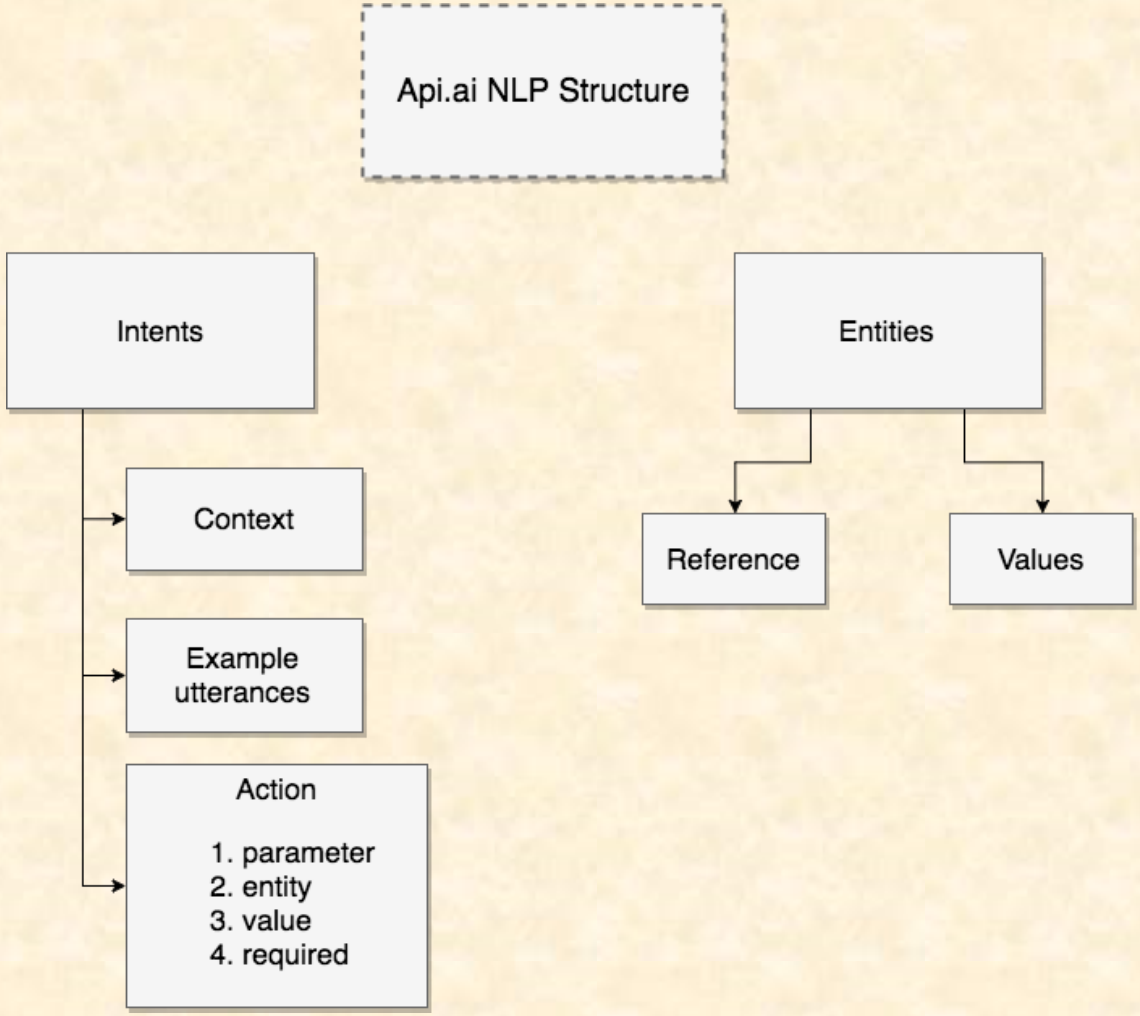
"Alexa, what time does MSUFCU open today?"

"Alexa, show me my last 5 transactions."

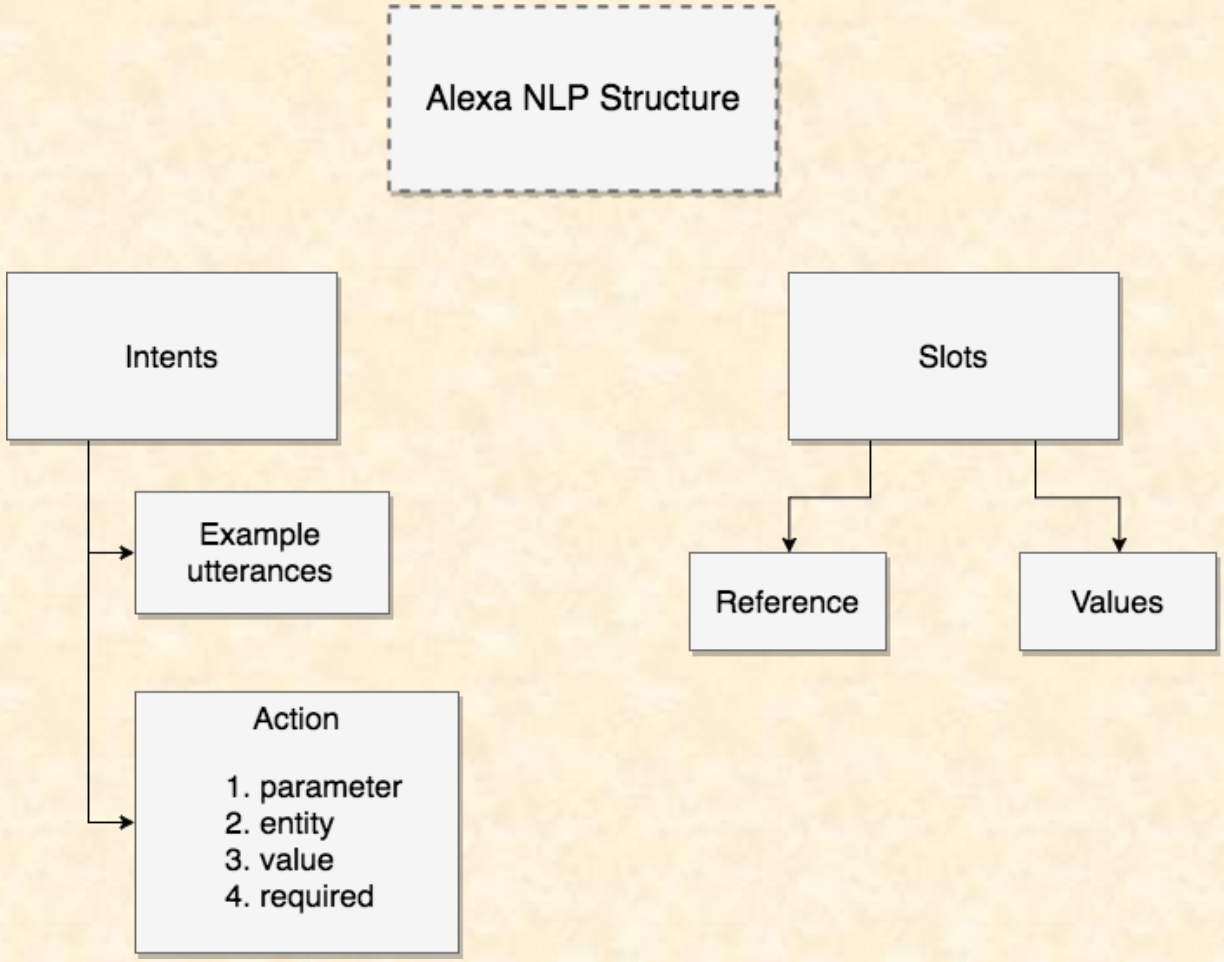
Technical Specifications

- API.AI: Google's NLP kit
  - Integrates with most platforms
  - Centralizes chatbot
- Web chat, FB Messenger, Twilio SMS, iOS app, Google Actions: integration with API.AI
- Amazon Alexa uses separate proprietary NLP platform
- Fulfillment/Webhook: Node.js app that handles routing and logical flow
- API: Node.js/Express app with a SQLite Database
- MSUFCU API: Allows access to customer data

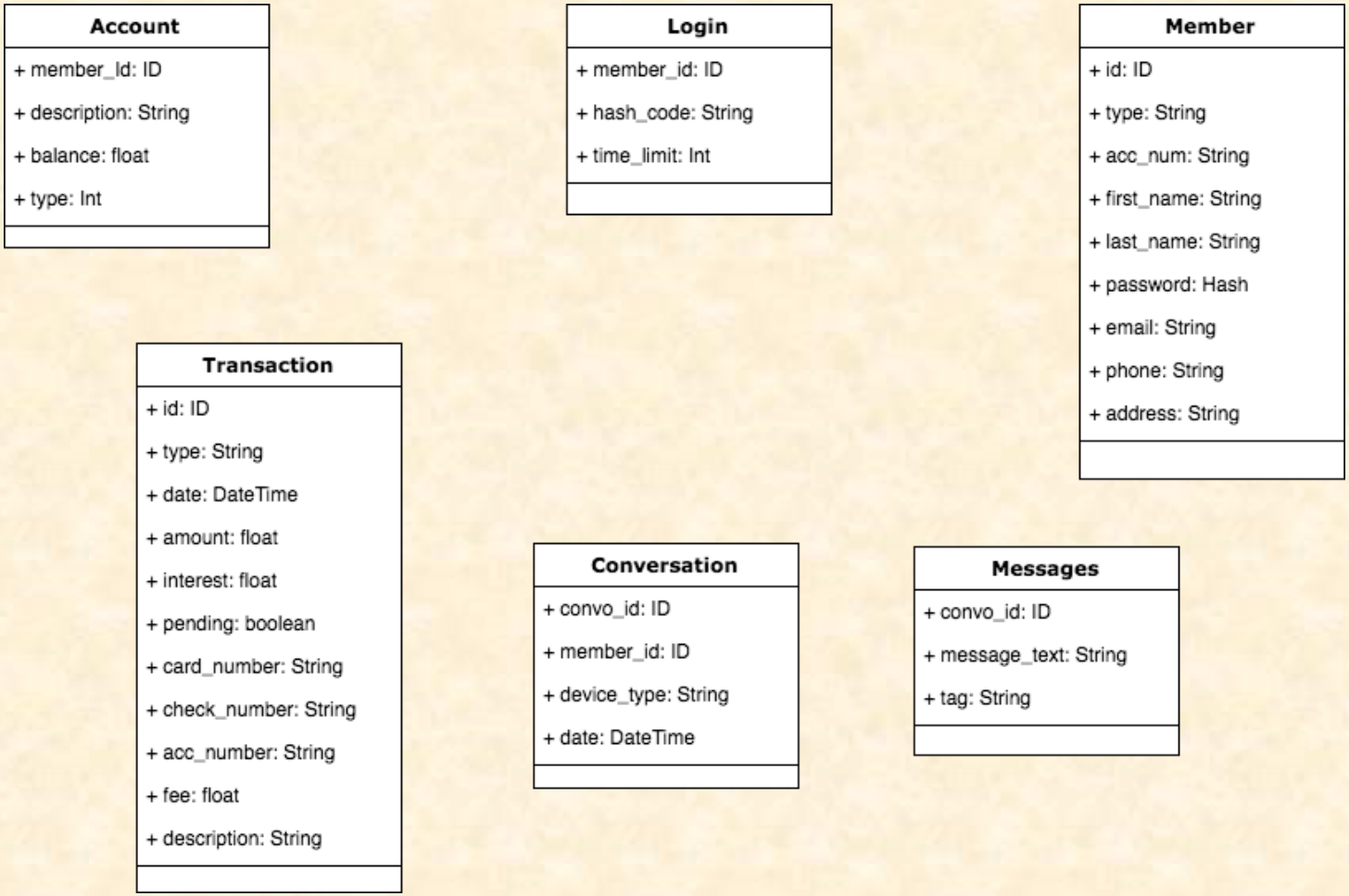
System Architecture: Api.ai



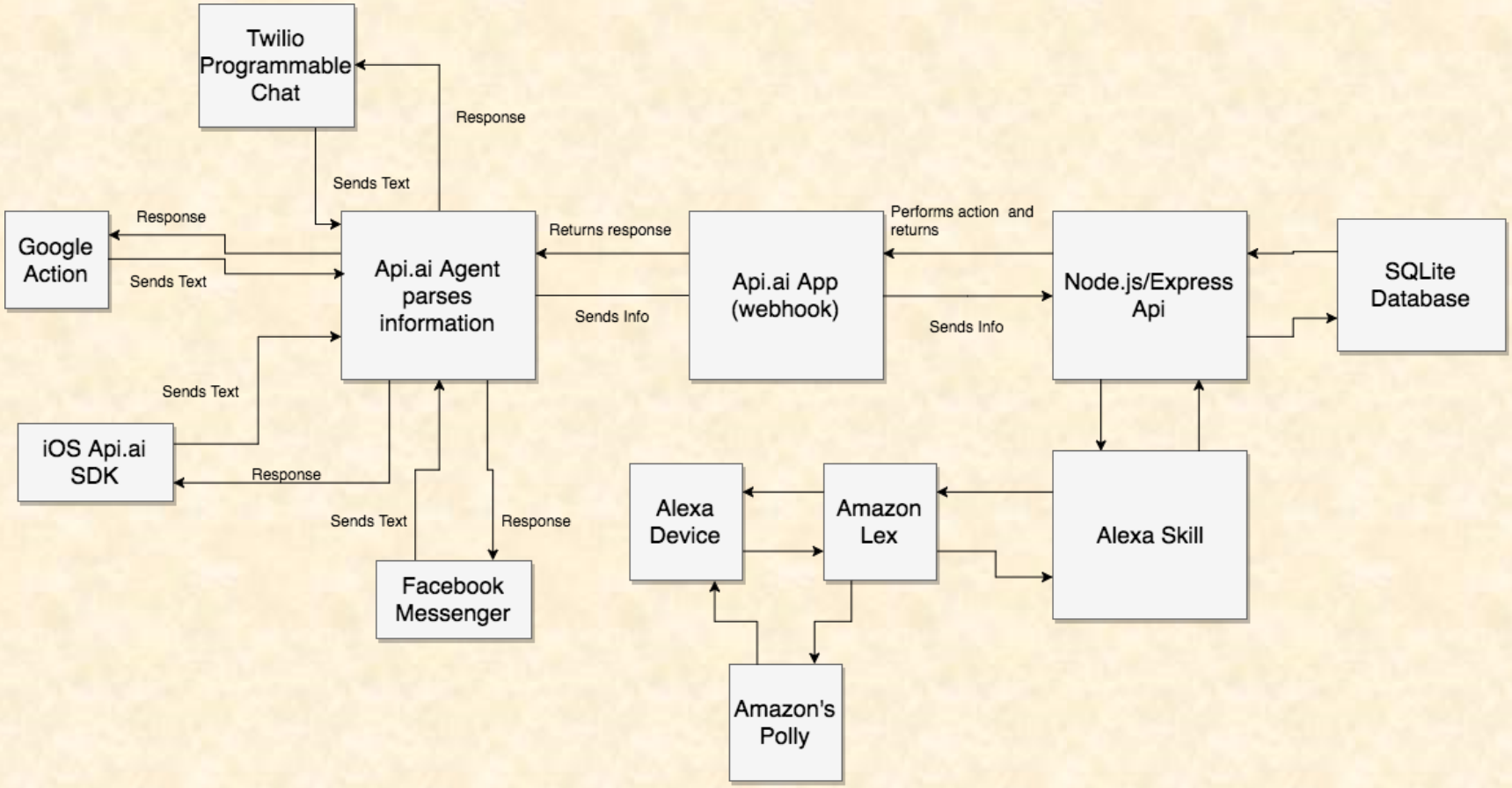
System Architecture: Alexa



System Architecture: Database



System Architecture: App



System Components

- Hardware Platforms
  - Mobile Phones (iOS app, SMS, Facebook Messenger, Google Assistant)
  - Desktop(Facebook Messenger and Web App)
  - Amazon Echo and Echo Tap/Google Home
- Software Platforms / Technologies
  - Android Studio/Xcode
  - API.AI, Amazon Lex/Poly
  - Webstorm

Testing

- Test conversation portion of chatbot by ensuring chatbot responds with context
- Testing logic flow of login to ensure it’s secure and session expires
- Load test Api.ai/API/Database with 3000 requests in a day
- Test chatbot to database through API connection to make sure that data is updated correctly in the database schema
- Testing Twilio from multiple phones to ensure it’s phone number agnostic
- Using Mocha and Chai for Unit Testing(JS Test framework)
- Custom Testing platform for automating voice testing to hit API

Risks

- Risk 1
  - Potential vulnerability of intercepting confidential data from database
  - Integrate SSL to create an encrypted tunnel between client and server
- Risk 2
  - Gaining access to an unauthorized bank account
  - Implement robust user access control to provide access to authorized users only
- Risk 3
  - Client wants apps on a variety of different platforms
  - Determined Google's API.AI is the best fit as it has integrations for multiple platforms. Prioritizing which platforms to focus on through user usage statistics
- Risk 4
  - NLP may not be able to understand what the user says due to accents and pitch
  - Custom testing platform to find words that are similar

Questions?