

INTERACTIVE TV DASHBOARD

Team: DEC1716

Client: UAVX LLC

Adviser: Swamy Ponpandi

Team Members:

Marco Restaino (Webmaster)

Jackie Larin (Communications Leader)

William Tangney (Team Leader)



Outline

- ❖ Motivation
- ❖ Problem Statement
- ❖ UAVX
- ❖ Functional Requirements
- ❖ Non-Functional Requirements
- ❖ Targeted Hardware Device
- ❖ Additional Device
- ❖ Technical Resources Utilized
- ❖ Implementation Details
- ❖ System Diagram
- ❖ Demo
- ❖ Design Challenges
- ❖ Design Changes/Previous Iterations
- ❖ Testing
- ❖ Conclusion/Future Work



Motivation

- ❖ According to a 2015 study regarding television statistics in U.S. households approximately 64% of households contain a television in the master bedroom of the home
- ❖ No other application like this presents the user's information with a television as the primary interface



Problem Statement

- ❖ This project was to create a dynamic and interactive TV dashboard on an HDMI dongle running an Android operating system
- ❖ It integrates a large list of API's to present items to the dashboard such as calendar events, weather, commute time, etc



UAVX

- ❖ UAVX LLC is a small startup that primarily works on drone components
- ❖ This capstone project idea was a proof of concept for them
- ❖ For more info visit www.uavx.com



Functional Requirements

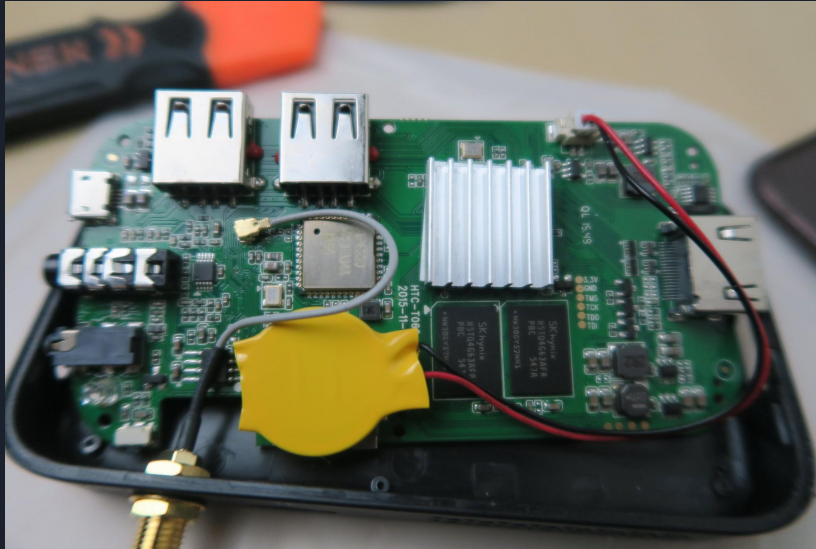
- ❖ Integrate multiple API components into single application
- ❖ Implement 3rd party APIs to deliver information to the user
- ❖ Keep all information services up to date automatically
- ❖ Integrate voice commands to request information from the dashboard interface



Non-Functional Requirements

- ❖ Clean and simple dashboard user interface
- ❖ Secure application
- ❖ Easy to use application
- ❖ Design to require minimal setup time

Targeted Hardware Device



- ❖ Android 5.1.1
- ❖ HDMI-CEC Support
- ❖ GPU supported by OpenCV
- ❖ Device targets TV and other large screen devices
- ❖ On board Wi-Fi
- ❖ Silent Operation
- ❖ Low power consumption

Additional Device



Amazon Alexa Dot

- ❖ This device, external from our system, enables the voice interaction functionalities.



Technical Resources Utilized

- ❖ Android Studio
- ❖ Bitbucket
- ❖ Google Maps, Email, and Calendar APIs
- ❖ Yahoo Weather API
- ❖ Alexa Skills Kit
- ❖ Amazon Web Services



Android Implementation Details

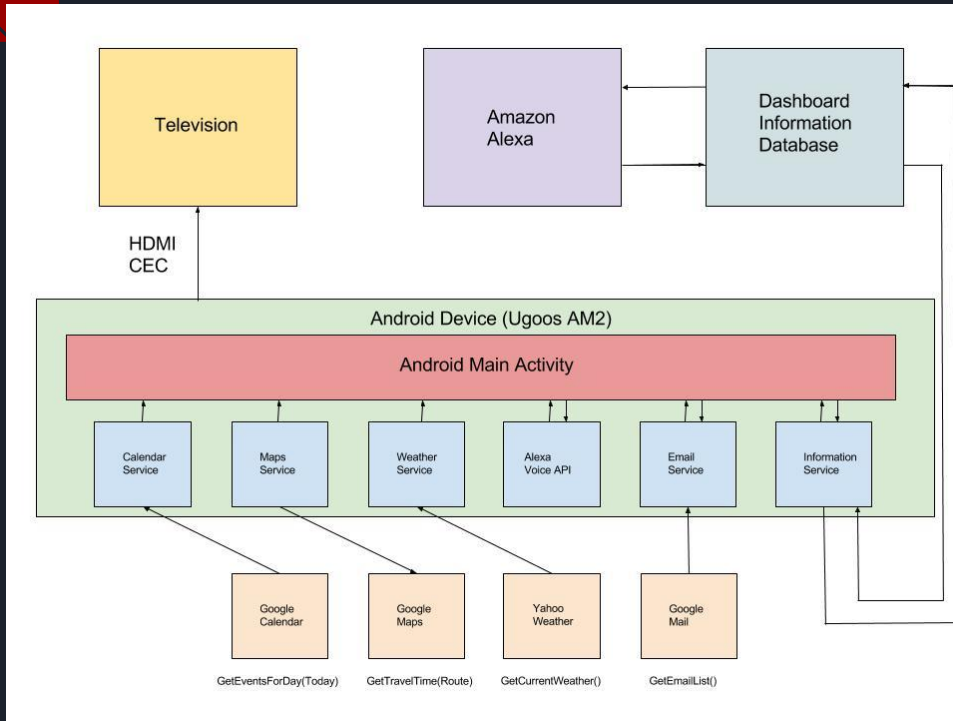
- ❖ Java
- ❖ Activities
- ❖ Core Classes
- ❖ Service Classes
- ❖ Provider Classes



Alexa Implementation Details

- ❖ Node.js and Java
- ❖ Alexa Skills Kit
- ❖ Alexa Web Services
 - AWS Lambda
 - AWS Cognito
 - AWS DynamoDB
 - AWS Android SDK

System Sketch



- ❖ `getEventsForDay()`
- ❖ `getTravelTimeforRoute()`
- ❖ `getCurrentWeather()`
- ❖ `getEmailList()`
- ❖ `updateInformation()`

Demo





Design Challenges

- ❖ Team Android Knowledge
- ❖ Team Alexa Knowledge
- ❖ User Interface Design Knowledge
- ❖ Hardware Limitations



Design Changes/Previous Iterations

- ❖ Direct interfacing with the Alexa
- ❖ SQL Server data services for Alexa
- ❖ Hand Tracking Functionality
- ❖ Google Weather API
- ❖ UI Element Presentation



Testing

- ❖ **Email Services and providers**

- Compare list of Emails to list of emails directly from 3rd party app. I.e. Compare gmail list to list from gmail app.

- ❖ **Route planning**

- Compare to other route planning services.

- ❖ **Weather**

- Compare to another weather service and compare weather data within a tolerance.

- ❖ **Calendar Events:**

- Add events to google calendar and make sure all events are accurately retrieved.



Conclusion/Future Work

- ❖ Customizable UI Elements
- ❖ Additional Alexa Commands
- ❖ Interface with other smart home type devices
- ❖ Move product closer to full home automation



Q&A



Thank You For
Your Time