




INTERACTIVE TV DASHBOARD

Marco Restaino (Webmaster)
Jackie Larin (Communications Leader)
William Tangney (Team Leader)



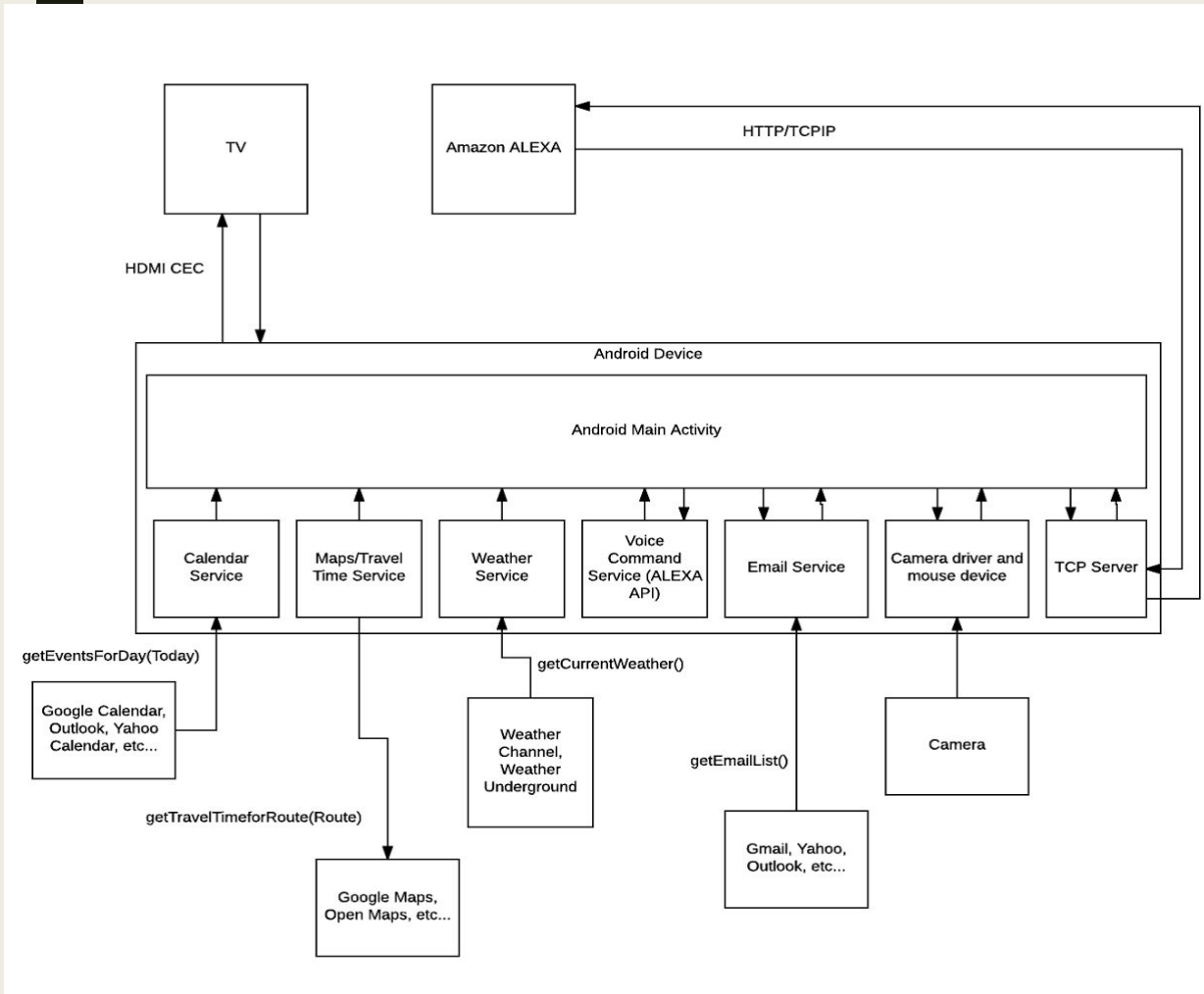
UAVX

- UAVX LLC is a fast-growing tech company located in Southern California. Their primary product is remote controls for drones. They are a small startup and this capstone project will be a proof of concept that they hope to be able to sell at some point in the future.
- For more info visit www.uavx.com

Problem Statement

- This project is to create a dynamic and interactive TV dashboard on an HDMI dongle running an Android operating system. We will incorporate the CEC functionality of HDMI to turn your tv on and alert you for the purposes of an alarm clock. It will integrate a large list of API's to present items to the dashboard such as calendar events, weather, commute time, etc.

Conceptual Sketch



- `getEventsForDay` - Sample events will be prepared in the calendar service and the list of events will be compared against the list from the calendar API.
- `getTravelTimeforRoute(start, finish)` - The calculated route will have to be checked against that of the API that is used to call or other APIs to validate its accuracy within a certain tolerance. The shortest route heuristics can sometimes produce different answers so it's best to compare within a tolerance.
- `getCurrentWeather()` - Check the current weather with calls from other weather reports. Compare within a tolerance. It can also be checked against a digital thermometer for temperature.
- `getEmailList()` - Check if the retrieved list of emails matches that of the user's email client.

Functional Requirements

- Integrate multiple API components into single application.
- Implement 3rd party APIs to deliver information to the user.
- Hand tracking control of the dashboard.
- Integrate voice commands to dashboard interface.

Non-Functional Requirements

- Clean and simple dashboard user interface.
- Secure application, impossible to exploit.
- Applications scales multiple types of Android devices and screen sizes.
- Avoid legal and licensing issues.

Interface Specifications

Good Morning Ivan.

5:10am

☀️ 61°

CNN China, Russia fear the US is boxing them in

abc Driver Avoided T Remote-Control

AUDIO ONLY **TUNE IN**

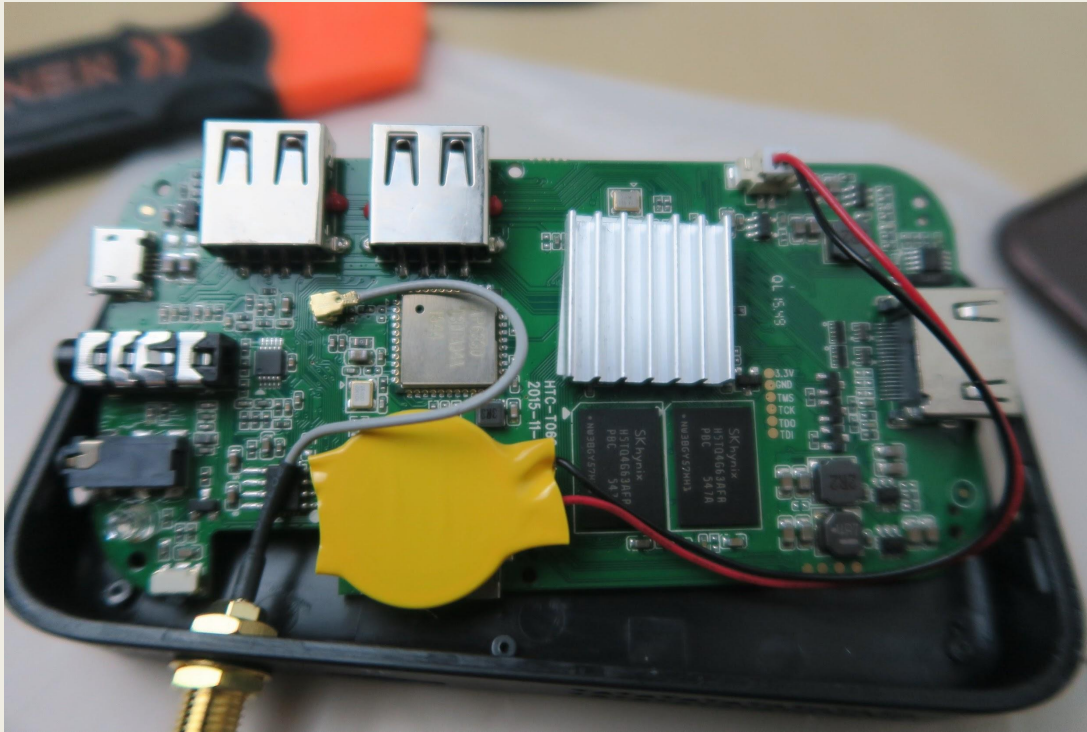
Here's how your day looks

- 5:30** PICK UP MOM FROM AIRPORT
📍 Los Angeles International
- 9:15** DENTIST APPOINTMENT
📍 402 N Larchmont Blvd
Los Angeles, CA 90004, USA
- 11:30** LUNCH WITH FAMILY
📍 Lemon Tree Kids & Family Restaurant
- 2:00** DRIVE JEANNA TO THE AIRPORT
📍 Los Angeles International
- 4:30**

Software Specifications

- The software we are using for this project is the Android OS, so any type of hardware we tested with so far has been loaded with that OS. To begin our project we started testing with Raspberry Pi's that we loaded Android onto. This was just a placeholder until we all decided on the hardware our final version will run on which we just received. The new hardware that our final product will run on is a Android Kitkat media box that has the CEC functionality of HDMI.

Target Device



- Android 5.1
- HDMI-CEC Support
- GPU supported by OpenCV
- Device targets TV and other large screen devices
- On board Wi-Fi
- Silent Operation
- Low power consumption

Software Design

- **Activities:** The main activity will have the GUI that interfaces with the user. It will call upon services to provide content.
- **Core Classes:** Contain common wrapper classes for objects that will be used throughout the application. E.g WeatherObject, EmailObject.
- **Service Classes:** These classes are Android services bound to this application that will call providers to provide the main activity with content. E.g EmailService, WeatherService, MapService.
- **Provider Classes:** Providers interface directly with 3rd party APIs and are called by services to provide content. The content provided will be one of the core classes for the service to process. GmailProvider, YahooWeatherProvider...

Project Milestones & Schedule

Feb - March

- Finish first draft of the Project Plan
- Finish our research
- Get the hardware needed for the project

March

- Finish first draft of the Design Document
- Start developing the services: weather, email, calendar, maps

April

- Prepare for Group Presentations
- Finish final draft of the Project Plan
- Finish final draft of the Design Document
- Continue to work on the services

Test Plan

- 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 heuristics. This can only be tested in a tolerance because brute force shortest path on a map takes a significantly long time.
- 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.

Prototype Implementations

- HDMI-CEC test app: This app controlled basic CEC commands and tested CEC functionality on home TV.
- General Android Experiments: Things to practice interaction with the user and how services and activities function together.

Current Project Status

- We have implemented the basic services: Weather, Email, Calendar, Maps
- We have received most of the hardware needed for the project - we are waiting a new monitor.
- We will continue to work on the basic services over the summer and be ready for the next phase of our project by August

Task Responsibility/Contributions

- William Tangney - Team Leader
 - *Created the template for the project and the backbone for the project*
 - *Working on the email service*
 - *Worked on the basic outline for all the services*
 - *Testing on the Intel Camera*
 - *Android expert*
- Marco Restaino - Webmaster
 - *Working on the Weather service*
 - *Kept our website up to date*
- Jackie Larin - Communications Leader
 - *Communicated with client and advisor*
 - *Working on the maps service*

Plan for Next Semester

August - September

- Develop GUI
- Fix any issues the application might be having
- Work on any missing service

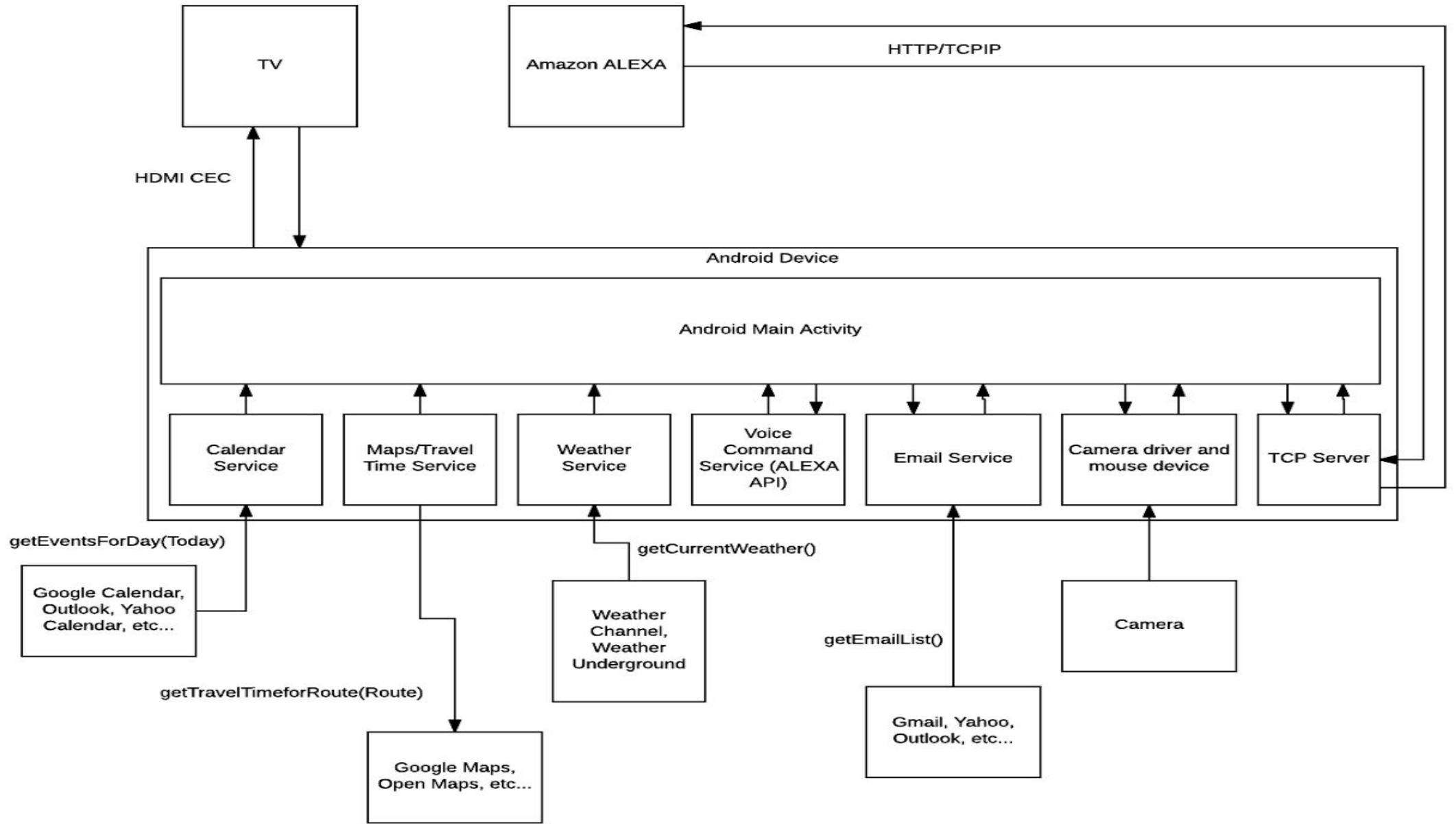
October - November

- Work on adding motion control - hand tracking
- Get different users to test the application
- Fix any issues the application might be having

December

- Display the project to the client
- Fix any issues the application might be having
- Turn over our project to the client





Ugoos AM2

