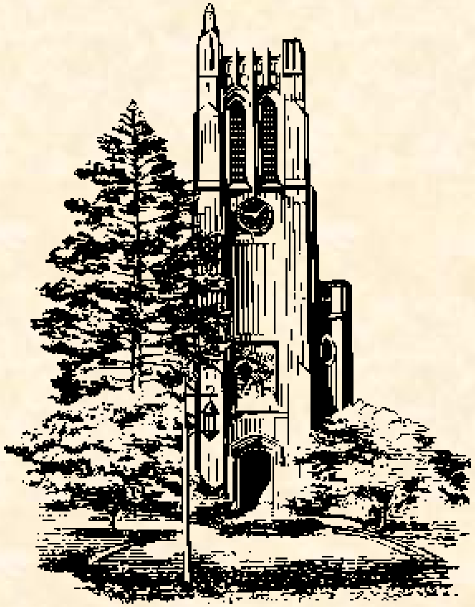
MICHIGAN STATE UNIVERSITY

Project Plan
Mobile User Generated Video Service



Team Motorola

CSE 498, Collaborative Design

Robert Gaul
Brandon Malicoat
Kurt Seippel
Jonathan Szynkowski

Department of Computer Science and Engineering
Michigan State University

Spring 2010

Project Overview

- Allow users to upload video via Android device
  - An iPhone application is already built but will require enhancements
- Associate friends via Social Network or either phone applications
- Friends can now watch the video on their STB
- Enhancements to architecture
  - Streaming video
  - Universal account and multiple providers
  - Better viewing session for the user

Functional Specifications

- Android application
  - Record video
  - Upload video to the social network
  - Add/Remove buddies
  - Associate/De-associate buddies with the video
- Services Proxy
  - “Universal” account
  - Multiple providers
    - Facebook, YouTube, etc

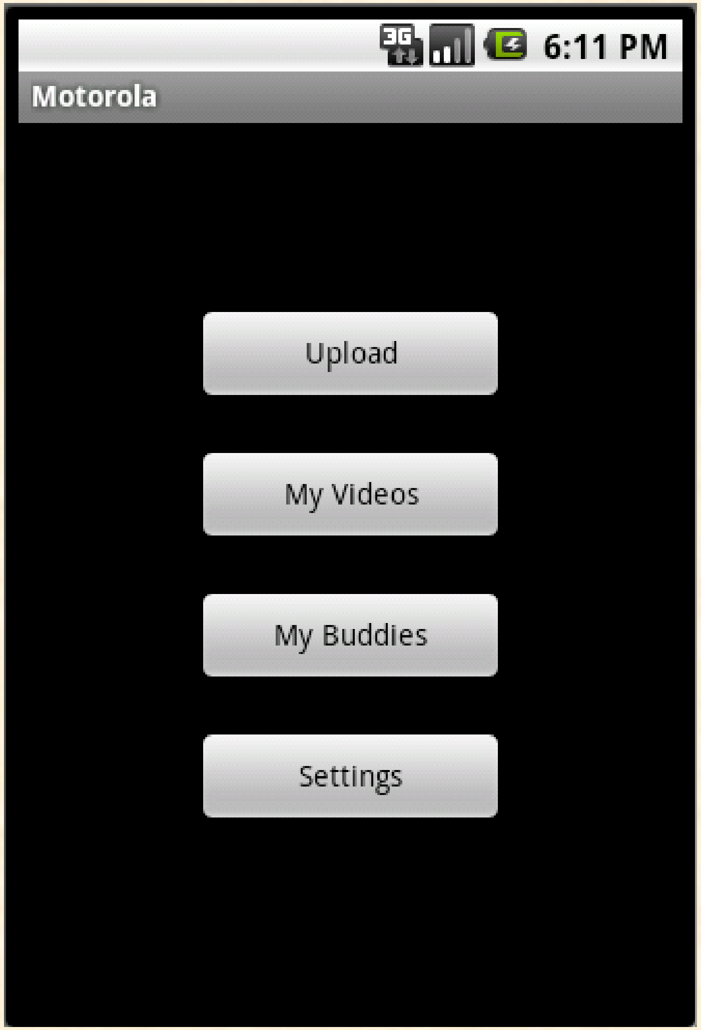
Functional Specifications

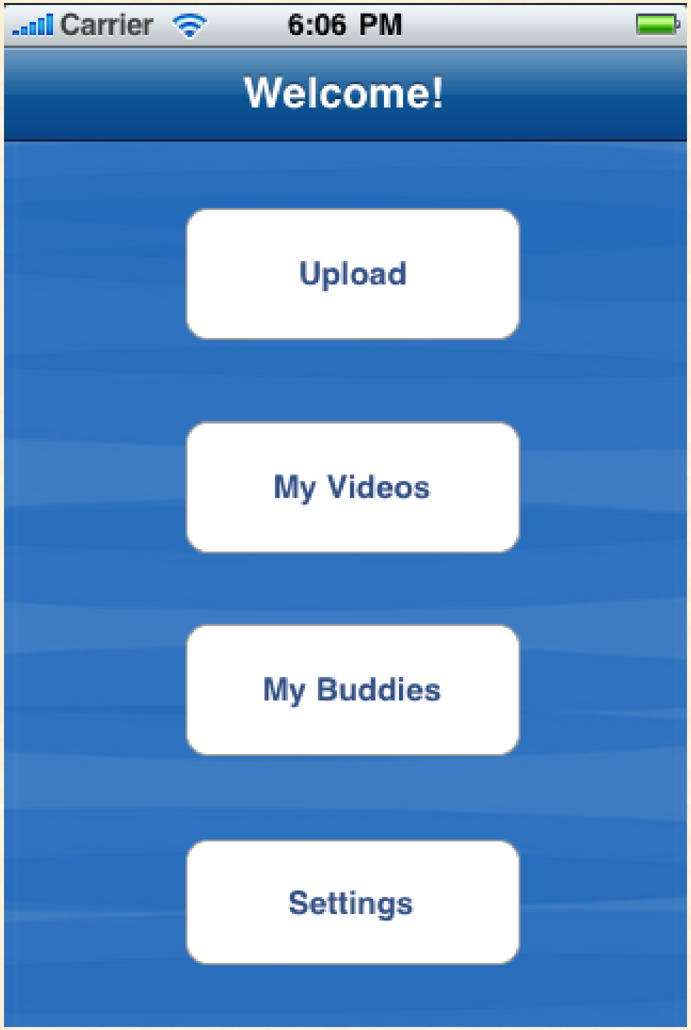
- Media Player
  - JVLC → VLC
  - Streaming
    - From social network to either the cable headend or the set top box
    - Do not download video to the VOD server
  - Tricks
    - Smooth the viewing experience

Design Specifications

- Android application
  - Similar look and feel as the iPhone app
- Services Proxy
  - Social network site = central hub
  - UI for adding providers and accounts
- Media player
  - Add VLC media player implementation

Screen Mockups

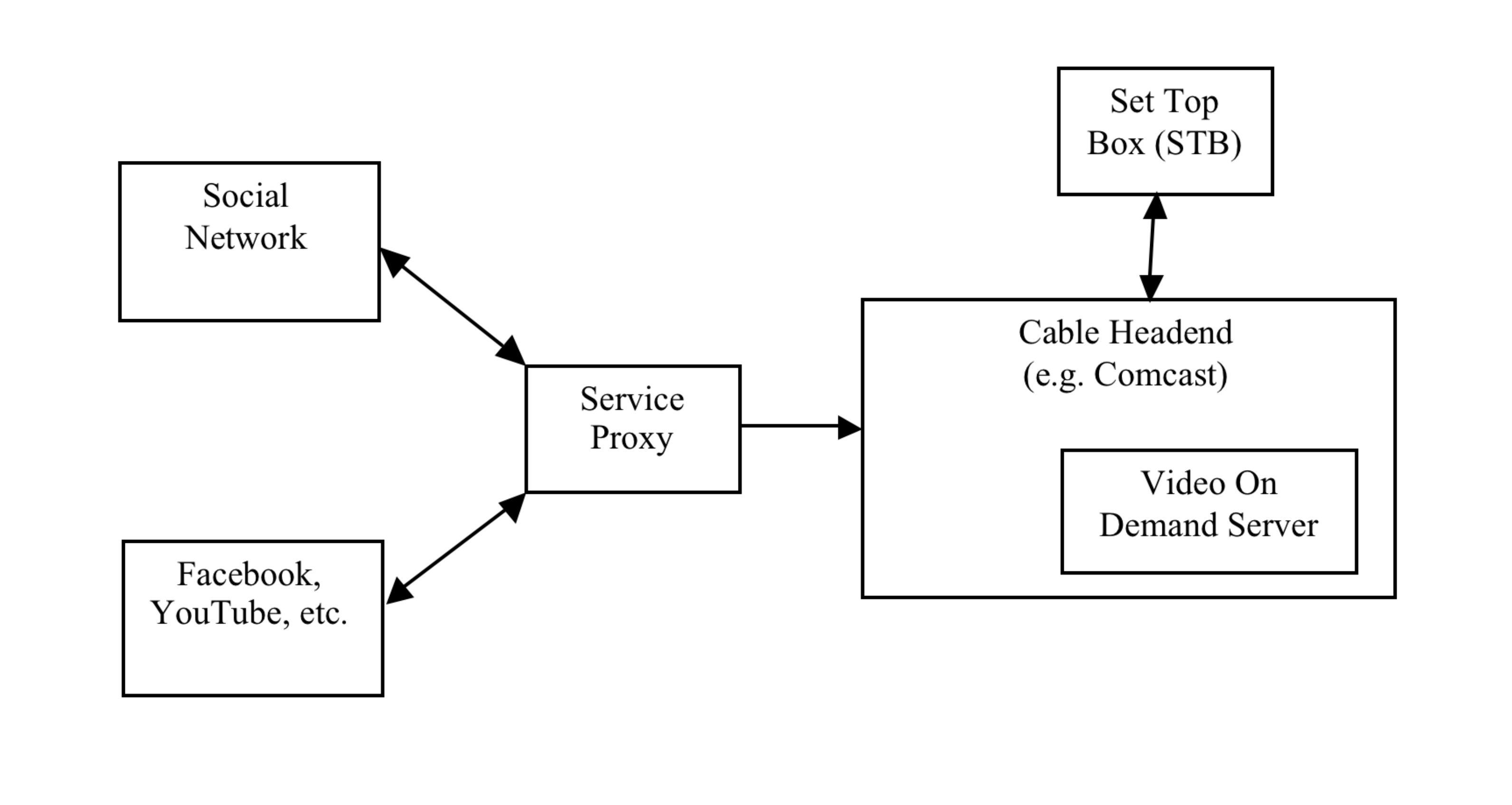




Technical Specifications

- Android application
  - Calls to RESTlet API's
    - Get list of videos
      - GET at /nss/{userID}
    - Upload video
      - POST at /nss/{userID}
- Services Proxy
  - Extend database to allow multiple STBs per user and multiple provider identities
- Media Player
  - Extend the Media Player interface to use VLC

Architecture Illustrated



System Components

- Hardware Platforms
  - Windows, Mac OS X, and Linux
  - Television with STB
  - iPhone and Android
- Software Platforms / Technologies
  - Eclipse with Android SDK
  - Xcode with iPhone SDK
  - Glassfish server
  - iBatis

Testing

- Android application
  - Make sure it is user friendly and intuitive
  - Mimic all functionality of existing iPhone app
- Services Proxy
  - Test UI in multiple browsers
  - Make sure all input is validated correctly
  - Ensure database stays in sync with all other databases
- Media player
  - Use multiple video types
  - Vary size of videos (including empty files)

Risks

- Android development
  - Application is near completion
  - No longer a priority risk
- Existing code base
  - Starting to understand a lot of the functionality thanks to Kabe
  - Still more to understand
- Understanding the entire system for new enhancements