MICHIGAN STATE
UNIVERSITY

Project Plan
Australis-Styled Widgets for Firefox

The Capstone Experience

Team Mozilla
Dan Poggi
David Thorpe
Eric Proper
Eric Slenk

Department of Computer Science and Engineering
Michigan State University

Fall 2013

From Students…
…to Professionals

Project Overview

- Three Australis-styled widgets
- Bugzilla
  - Interfaces with Bugzilla (Bug Tracking System)
  - Alternative to Bugzilla User Dashboard (Web Page)
- Music
  - Local music player
  - Basic music playback functionality
- Weather
  - Displays basic, current weather information
  - Links to more complete forecast site

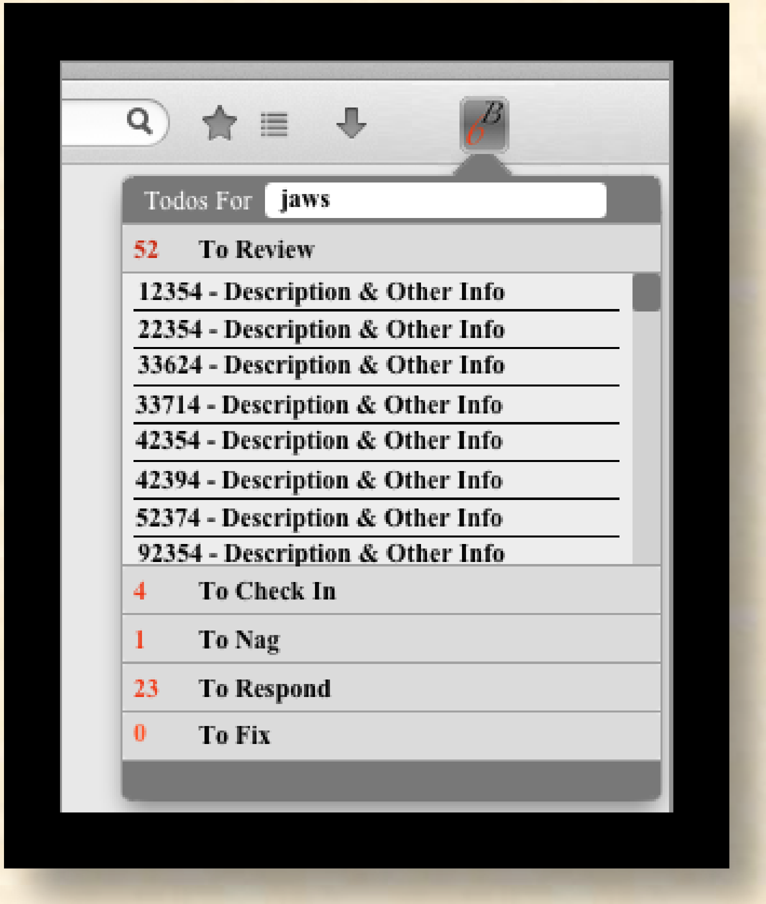
Functional Specifications

- Bugzilla
  - Ability to read number of new bugs from chrome icon
  - Enter and save user to widget
  - Bug categories, expand to show individual bug info
  - Bug names linked to Bugzilla web page
- Music
  - Play music from user-selected local directory
  - Playback controls
  - Allow users to choose songs to play from add-on
- Weather
  - Ability to read current weather from chrome icon
  - Clicking on icon reveals all cities weather
  - Add/delete cities to main panel
  - Click to see more detailed forecast on external web page

Design Specifications

- Bugzilla
  - Live icon displaying the number of new bugs for the user
  - Input field to specify the user’s nickname
  - View that displays a list of bug categories
    - Categories expand to show individual bugs
    - Clicking a bug name will open a new tab in the browser
- Music
  - Music player, follows basic mini player design
  - Circular ordered list of user’s songs
  - Settings menu for setting the music directory
- Weather
  - Live icon displaying the current locations weather
  - Main panel includes weather for all user-added cities
  - View for user customization (i.e. adding/deleting cities)

Screen Mockup: Bugzilla Interface



Todos For jaws

52 To Review

12354 - Description & Other Info

22354 - Description & Other Info

33624 - Description & Other Info

33714 - Description & Other Info

42354 - Description & Other Info

42394 - Description & Other Info

52374 - Description & Other Info

92354 - Description & Other Info

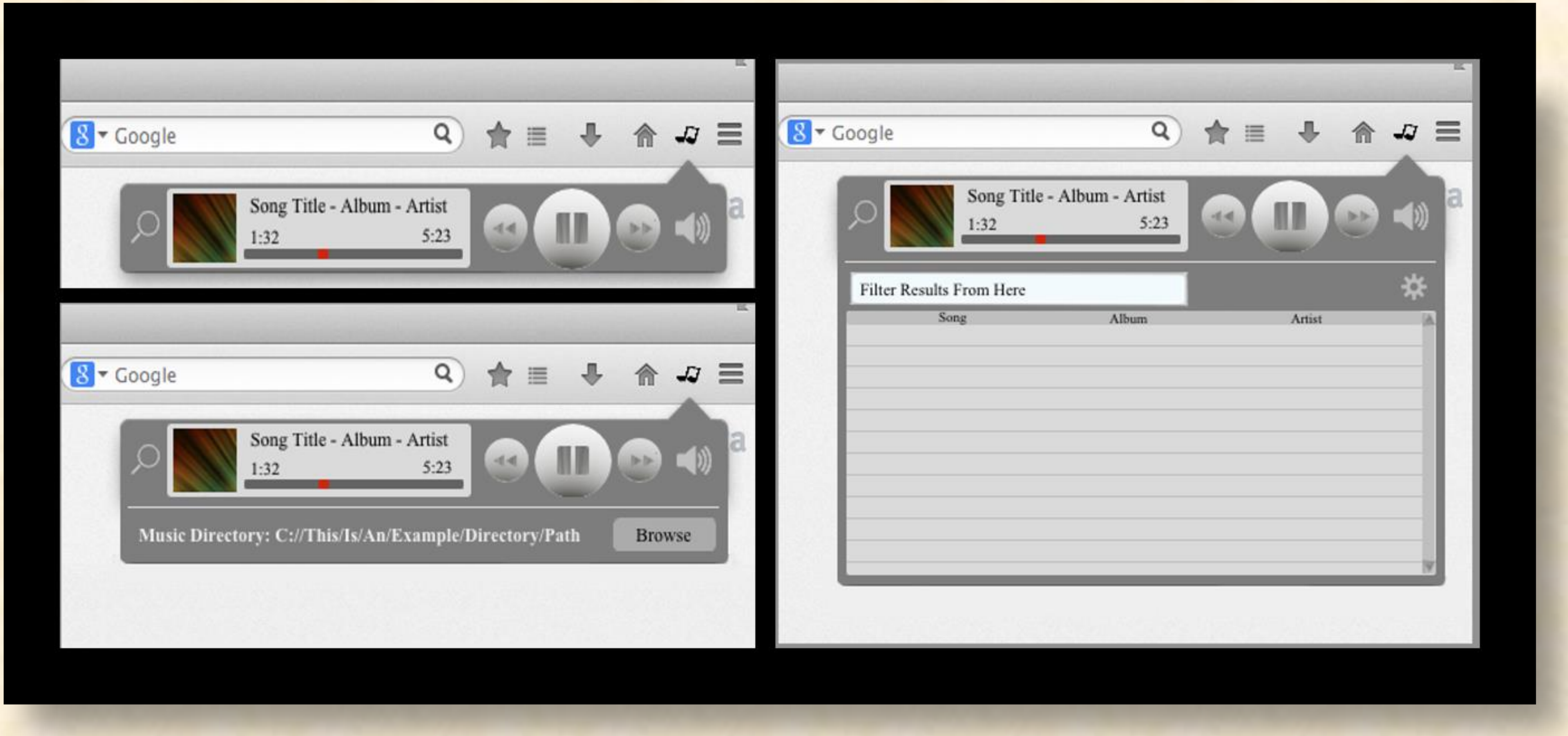
4 To Check In

1 To Nag

23 To Respond

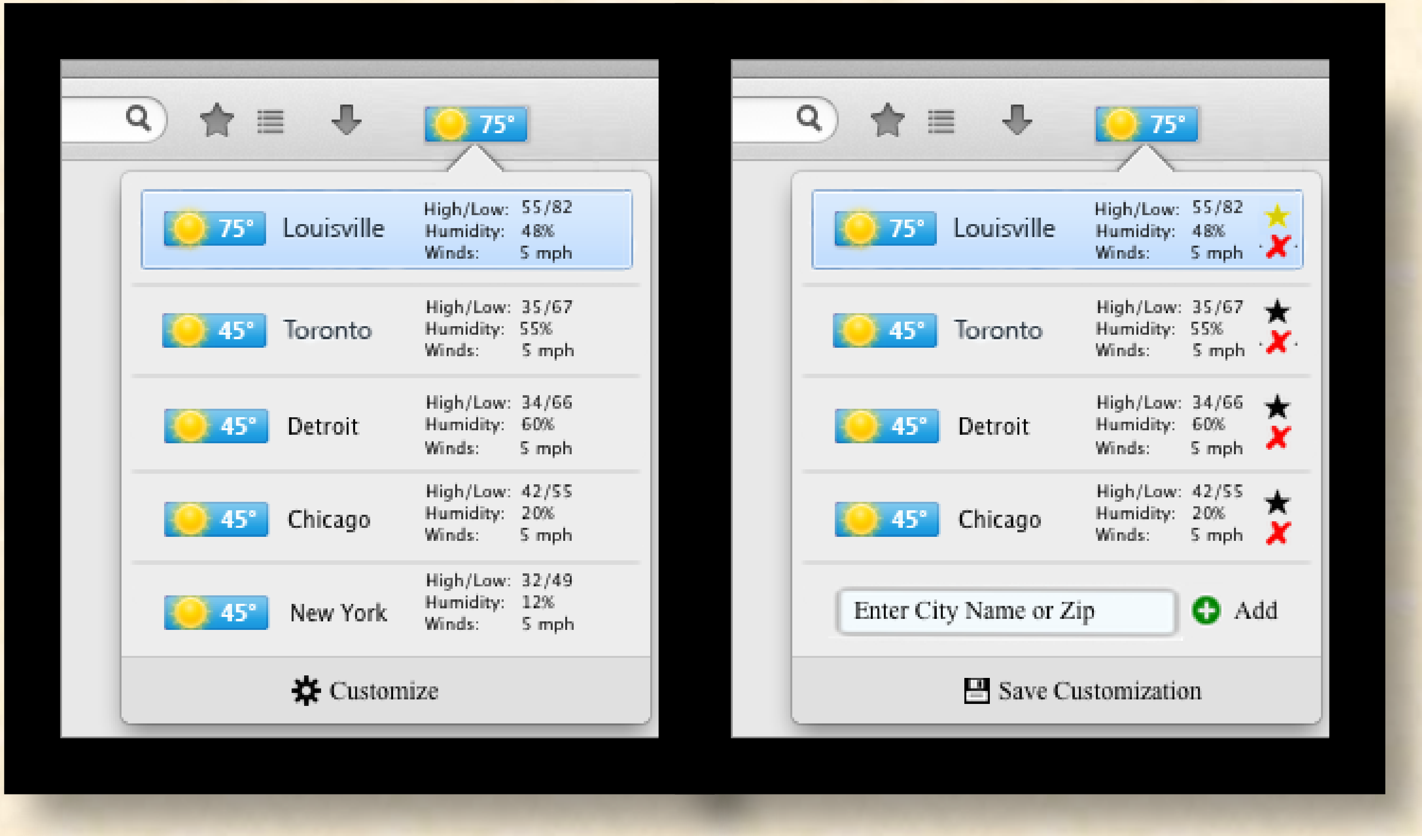
0 To Fix

Screen Mockup: Music Interface



Google
Song Title - Album - Artist
1:32
5:23
Google
Song Title - Album - Artist
1:32
5:23
Music Directory: C://This/Is/An/Example/Directory/Path
Browse
Google
Song Title - Album - Artist
1:32
5:23
Filter Results From Here
Song
Album
Artist

Screen Mockup: Weather Interface



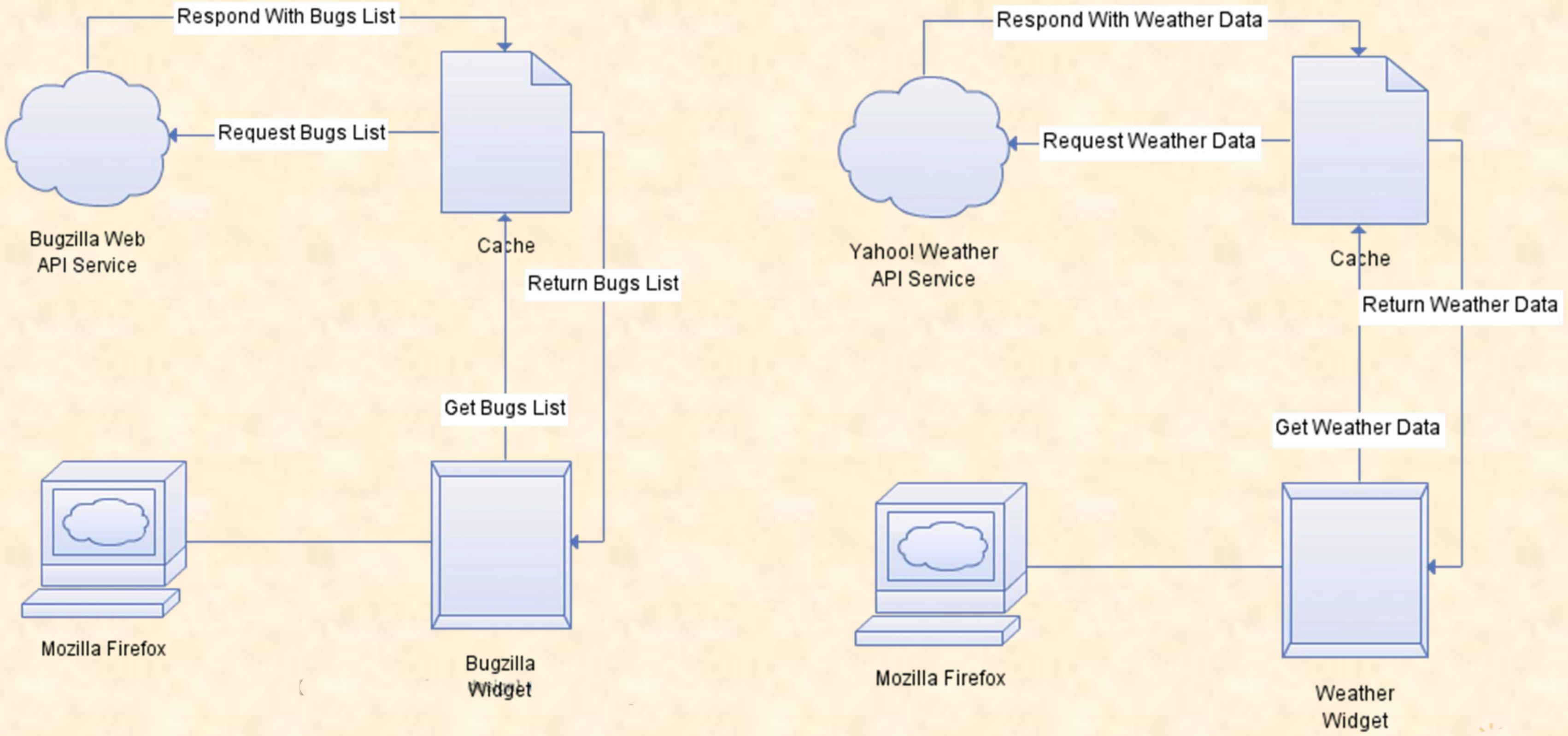
75°	Louisville	High/Low: 55/82 Humidity: 48% Winds: 5 mph
45°	Toronto	High/Low: 35/67 Humidity: 55% Winds: 5 mph
45°	Detroit	High/Low: 34/66 Humidity: 60% Winds: 5 mph
45°	Chicago	High/Low: 42/55 Humidity: 20% Winds: 5 mph
45°	New York	High/Low: 32/49 Humidity: 12% Winds: 5 mph

75°	Louisville	High/Low: 55/82 Humidity: 48% Winds: 5 mph	★ ✗
45°	Toronto	High/Low: 35/67 Humidity: 55% Winds: 5 mph	★ ✗
45°	Detroit	High/Low: 34/66 Humidity: 60% Winds: 5 mph	★ ✗
45°	Chicago	High/Low: 42/55 Humidity: 20% Winds: 5 mph	★ ✗

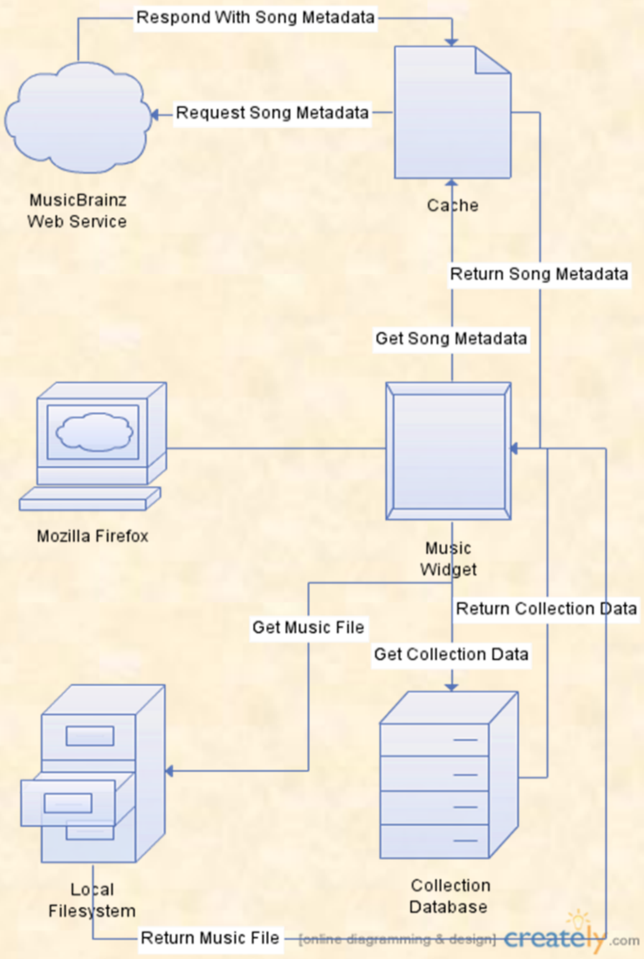
Technical Specifications

- General
  - XUL, HTML, JavaScript, CSS
  - Firefox Chrome API
- Bugzilla
  - Bugzilla REST API
  - Cache of Bugzilla bugs
- Music
  - MusicBrainz
  - Collection Database (Mozilla’s IndexedDB API)
  - Firefox <audio> API
  - Local file system
- Weather
  - Yahoo! Weather API
  - Cache of Yahoo! Weather data

System Architecture: Bugzilla/Weather



System Architecture: Music



Property	Type	Description
File	String	Path to the song's file on the user's local file system.
Title	String	The name of the song.
Artist	String	The name of the artist who produced the song.
Album Artist	String	The name of the artist who produced the album.
Album	String	The name of the album on which this song appears.
Genre	String	The genre of the song.
Composer	String	The name of the composer(s) of the song.
Track	Number	The position of the song in the album.

System Components

- Hardware Platforms
  - x86, x86-64
  - Desktops and laptops, no mobile (Android) support
- Software Platforms / Technologies
  - Windows (XP through 8), OS X (10.6-10.8), Linux (Ubuntu Unity & Gnome 3)
  - Firefox 26 (Australis)/SpiderMonkey JavaScript engine

Testing

- Automated Testing
  - Unit Testing
    - Mochitest Framework – JavaScript functionality
    - Mozmill – User interface functionality
  - Integration Testing
    - Integrate UI code with business logic code.
    - API integration (add-on code with web services)
    - Add-on functions properly within the browser
- User Experience
  - Community surveyed about their experience
  - Concentration on user interface, but includes functionality

Risks

- Australis-Styled Widgets
  - No official documentation for creating widgets using Australis widget APIs.
  - Client contacts have written much of the relevant C++ code. The client contacts have highlighted areas of the source code to review in order to understand new functionality.
- ECMAScript 6
  - Firefox's browser chrome is written in ECMAScript 6. The ECMAScript 6 standard is a work in progress and adds significant new syntax to the language.
  - Mozilla maintains a document describing all new ECMAScript 6 syntax and libraries which are implemented in the most recent version of Firefox.

Risks

- Third-Party APIs
  - Each add-on depends on an external third-party web service over which the developers have no control. Add-on functionality may be impacted by API changes, service outages, etc.
  - Mock responses will be written for all APIs so developers can develop and test against the API.
- Cross-Platform Music Playback
  - Cross-platform music playback will be handled using the Web Audio API. Some common music file types, such as MP3 or AAC, do not have the same support on all platforms.
  - The music player will use a pure JavaScript library called aurora.js. aurora.js decodes and plays all formats not natively supported by the browser.