

CptS 323 Software Design

Project Management and Milestones

The project has several milestones with their respectively due date. At each milestone, you may be asked to demonstrate your progress by a set of software artifacts defined at that milestone. At the end of each milestone, we will decide objectives for the next milestone. The final product is due in the final week of lecture.

Milestone Objectives

You'll do your project in phases, with the following milestones:

Preliminary

Due date: February 1st.

Objectives:

- Team selection: Select your partners to form a team of three. If you don't have a team by the due day, I will assign you to a team.
 - Select a team liaison and report the name to the TA.
 - Pick a name for your team.
- Setup your **eeecs.wsu** email. (How to setup your eeecs email?)
- Login to the EECS GitHub server at <https://github.eecs.wsu.edu> using your EECS account credentials.
 - Use the repository under organization "Cpts323Spring2013".
 - Get familiar with the EECSGitHub server.
- Establish the development environment. Install necessary tools.
 - Decide your choice of programming language: Java or C#.
 - Install Visual Studio for C# (from WSU Tech Store, <http://infotech.wsu.edu/helpdesk/Software.aspx>) or Netbeans IDE (<http://netbeans.org/>) for Java.
 - Get familiar with RSS feeds and RSS formats (RSS 1.0, RSS 2.0 and Atom 1.0)
 - Download and install Google Earth Plug-in (GE plug-in in short) and read the Google Maps/Google Earth APIs Terms of Service. Look for example codes that embed Google Earth Plug-in in standalone C# or Java applications.
 - A sample code (see the link below) for embedding Google Earth plug-in in a C# forms application is available with GNU Lesser General Public License. Download and compile the source and understand how the GE plug-in is integrated into the C# managed code. Explore the functionality and the control features that the control library provides.
 - **winforms-geplugin-control-library** : A library of controls to work with the Google Earth Plugin API in managed code.
<http://code.google.com/p/winforms-geplugin-control-library/>

Milestone-1

Due date: February 18.

Deliverables: underlined items are deliverables.

Objectives:

- **Communication activities:**
 - Develop “requirement specification”
 - Provide use cases
 - Provide itemized requirements
 - Template for the “Requirements Specification Document” will be posted on ANGEL next week.
 - Deliver a good quality document: completeness, feasibility, etc
- **Planning activities:**
 - Make an initial version of project schedule.
 - Identify the main tasks of the project (The complete set of milestone steps for implementing your project will be available next week. You need to create a more detailed listing of tasks based on the given milestones.)
 - Provide a Gantt chart to identify your work plan.
 - In the Gantt chart you don’t need to partition responsibilities to team members yet. Later you will revise your Gantt chart and assign a team member to each task.
- **Construction activities:**
 - Start to develop the “Main interface”.
 - Add menu items and implement the following functionality:
 - Add RSS subscription (get RSS feed URL from user, add an item for it in the RSS feed list – you don’t need to fetch and display RSS feeds yet)
 - Rename existing subscription,
 - Add new channel, rename channel, delete channel,
 - Organize RSS feeds into groups (add/delete RSS feeds to/from channels)
 - Write the code to parse RSS feeds (at least finish parsing RSS 2.0 feeds. Your final software should support RSS 1.0, 2.0 and Atom 1.0).

Milestone 2

Due date: March 8.

Deliverables: underlined items are deliverables.

Objectives:

- **Communication activities:**
 - Develop “Requirements Analysis Document”
 - Template for the “ Requirements Analysis Document” will be posted on ANGEL.
 - Deliver a good quality document: completeness, feasibility, etc

- **Planning activities:**
 - Refine your Gantt chart, add new tasks if needed, assign a team member to each task.
- **Construction activities:**
 - Fetch data from a (subscribed) RSS feed.
 - List the RSS items from an RSS feed. Display the following for each RSS item: item title, link to the original article, publication date of the original article, and item description (displaying the description is optional). You may limit the number of items you list from each feed. For example, list only the most recent 20 items from each feed. User may provide the number of items to be listed (optional).
 - Display the full article (when the user clicks on an article link). Requires rendering web pages.
 - The user sets the “update period” for the feeds (i.e. how often will the feed data be refreshed for the subscribed RSS feeds). You may also require a different “update period” for each individual feed (optional). Use a default update period if user doesn’t set one.
 - Refresh the RSS feed data for each subscribed RSS feed, according to the user defined “update period”.
 - Upload your current code on GitHub before the deadline.

Milestone 3

Due date: March 25.

Deliverables: underlined items are deliverables.

Objectives:

- **Communication activities:**
 - Develop “Software Design Document”
 - Template for the “Software Design Document” will be posted on ANGEL.
 - Deliver a good quality document: completeness, feasibility, etc
- **Planning activities:** TBA
- **Construction activities:**
 - Parse other two RSS formats (RSS 1.0 and Atom 1.0)
 - Keep track of the read and unread articles. Use a visual indication that shows read and unread articles. Users should be able to list the articles that have not yet been read yet.
 - Save the current configuration and user preferences to a config file. You may save the following: the collection of feeds, their (user given) names, feed update periods, organization into channels
 - Load a previously saved config file.
 - Build a local database to store all state names/abbreviations, and city names along with their coordinates in the United States. You need to create this database once.
 - Cover United States map with bounding boxes and query the Geonames webservice to obtain location names. Insert keywords and locations (latitude/longitude) into the database.
 - Upload your current code on GitHub before the deadline.

Sample code (in C#) for handling database operations (create database, insert/search to/from database) will be provided by instructor.

Milestone 4

Due date: April 8.

Deliverables: underlined items are deliverables.

Objectives:

- **Communication activities:** NA
- **Planning activities:** TBA
- **Construction activities:**
 - Search the local location database for the words that appear in the titles and descriptions of the RSS feed items. If one or more words from the RSS feed item exist in the location database, then retrieve the associated location(s). If none exists, then no location can be associated with that feed item.
 - Embed the Google Earth Plug-in in your application. You may use the winforms-geplugin-control-library, available at <http://code.google.com/p/winforms-geplugin-control-library/>. This is a library of controls to work with the Google Earth Plugin API in managed code. It is available with GNU Lesser General Public License.
 - **Upload your current code on GitHub before the deadline.**

Milestone 5

Due date: April 22.

Deliverables: underlined items are deliverables.

Objectives:

- **Communication activities:**
 - Develop “**Design Patterns Specification**” (Template is available on ANGEL)
 - Develop “**Report on Version Control and Issue Tracking**”(cancelled)
- **Planning activities:** TBA
- **Construction activities:**
 - Display the locatable RSS feed items on the map interface. Have an icon at the associated location for each of those feed items. The locatable RSS feed items include those that have associated location information (latitude/longitude).
 - Open a pop-up message on the map interface where you display the title and description of the article, along with a link to the article. The user should be able to view the article in the “main interface” by clicking that link.
 - Implement time filtering (optional) Both interfaces (main and map interface) should allow the user to view the articles on certain dates and/or time only. For example, if the

user wants to see the most recent news articles posted within the last 2 hours, the main interface should filter the RSS feeds accordingly. Similarly on the map interface, only the icons for the articles posted within the last 2 hours should be displayed.

- **Upload your current code on GitHub before the deadline.**

Product Release and Presentations

Due date: April 22-24.

Objectives:

- **Upload your final code on GitHub before April 24th.**
- **Communication activities:**
 - **Schedule a demonstration with the TA**
 - **~~Present your project in class (cancelled)~~**