Software Requirements Specification
for the
Rotation Curve Modeler (RoCM)

Version 2.0

Robert Moss, Alex Clement
Wentworth Institute of Technology

June 13, 2014

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	**Printed Name**	**Title**	**Date**

Contents

1 Introduction

1.1 Purpose

The Software Requirement Specification for the Rotation Curve Modeler will explain in detail necessary features that the client purposes and the developers provide. Astrophysicists will be the main target for this web application. It will expedite the process of accessing galactic data (from the project Scholarly Observed Celestial Measurements) and provide a tool to model galaxies using many different theories to solve the rotation curve problem.

1.2 Scope

1. Rotation Curve Modeler
   (a) Interact with SOCM (Scholarly Observed Celestial Measurements)
      i. Use the repository of observable galactic data to model hundreds/thousands of different galaxies.
   (b) Interact with RoCS (Rotation Curve Simulation)
      i. RoCS visualizes the spin of star clusters around the center of a galaxy.
      ii. Include scale and legend for the visualization.
   (c) Allow users to locally import and run their own model (via JavaScript code)
      i. Following the defined v(R) input/output standard.
      ii. Observable galactic parameters from SOCM will be available as constants.
      iii. User defined constants will need to be implemented in the user's function.
   (d) Locally import LaTeX equation for each model (optional)
      i. The user can import their own LaTeX equation to be displayed during the data plotting.
      ii. Aids in understanding the behavior of each parameter.
   (e) Dynamic parameter sliders
      i. For every parameter in the individual models, a dynamic slider with user defined ranges can be created.
2. rotationcurve.com / rotationcurve.org
   (a) The website will host the RoCM, SOCM, and ROCS components.
      i. This website, hosted on WIT servers, will allow astrophysicists to access a large database of observed galactic data (via SOCM), plot their own rotation curve models (via RoCM), and simulate their rotation curves (via RoCS).

1.3 Definitions, Acronyms, and Abbreviations

- Application Specific Definitions
  - RoCM - Rotation Curve Modeler
  - SOCM - Scholarly Observed Celestial Measurements
  - RoCS - Rotation Curve Simulator

- Industry Definitions
  - WIT - Wentworth Institute of Technology
  - SRS - Software Requirements Specification
  - JavaScript - A web based programming language.
  - D3 - Data Driven Documents: A JavaScript library for data visualization.
  - Ruby on Rails - A web development framework written in the Ruby programming language.
  - JQuery - A JavaScript library for easy UI development.
  - LaTeX - A document preparation system used widely throughout science and mathematics.
  - SVG - Scalable Vector Graphics: A loss-less graphics format.
- Technical Definitions
  - MoND - Modification of Newtonian Dynamics
  - TeVeS - Tensor-vector-scalar gravity
  - MATLAB - A mathematical programming language (MATrix LABoratory).
  - Mathematica - A mathematical programming language.
  - DB - Database
  - UI - User Interface
  - GUI - Graphical User Interface
  - HTML - HyperText Markup Language
  - div - HTML tag to define a division in a document
  - DOM - Document Object Model. A convention for representing and interacting with objects in HTML.
  - API - Application Programming Interface
  - km/s - Kilometers Per Second
  - kpc - Kiloparsecs

1.4 References

The list of references below are software documentation that we will be using:

1. Data Driven Documents (D3): http://d3js.org/
2. JQuery documentation: http://api.jquery.com/
3. JQuery UI documentation: http://api.jqueryui.com/

1.5 Overview

The rest of the SRS will contain:

1. Specific features RoCM and the rotationcurve.org website
2. System Requirements
3. Design Constraints for the application.
4. Risks within the scope.
5. Any additional information about the development of the application.

2 General Description

This section will explicitly lay out the product's functionality and constraints as well as compare it to existing products. Possible set backs will be discussed. Details about each requirement will be laid out in Section 3: Specific Requirements.

2.1 Product Perspective

Astrophysicists will need to model galaxies in programs like MATLAB or Mathematica, but there doesn't exist a singular tool to expedite this process in a universal format. Currently, scholars must sift through peer-reviewed articles and gather data one-by-one. rotationcurve.org will help generalize the work being done on galaxy research and provide one central repository for researchers.

RoCM will serve as a visual modeler for our collections of data and contemporary theories within astrophysics. With observable data as the input (via SOCM), any galaxy can be imported into the tool. The tool will plot obervational data and multiple galactic models together as a graph, as well as enable users to import their own galatic models to test against existing theories. Sliders will allow users to control parameters within their models with realtime visual feedback in the generated graph.

2.2 Product Functions

There will be many available functions of RoCM users can utilize. The modular design of the software will allow for additional functions to be easily implemented.

1. A rotation curve plotting tool that overlays different models of the galaxy on top of the observational data. Gives the user the ability to select which curve to plot based on the models available. Value ranged sliders that can update each individual parameter within specific galactic models. Each slider can be dynamically created with user defined ranges. This enables the user to visualize the behavior of each parameter within the each model. Allows for the testing of uncertainty within the galactic parameters.

2. The rotationcurve.org website will provide a web interface to the RoCM, SOCM, and RoCS modules. The website back-end will host galaxy research data, namely through SOCM, which will also provide API endpoints for ease of access and extensibility. rotationcurve.org will provide an 'About' section that explains the purpose of the website, and provides the email addresses to contact the contributors.

2.3 User Characteristics

The rotationcurve.org website and its constituent modules will be targeted at researching astrophysicists in search of peer-reviewed galaxy data. Users will also benefit from the visualization features for their galactic rotation curve research.

2.4 General Constraints

The D3 JavaScript library provides a fast an easy way to manipulate data and generate graphs in real time. Thus, it became our best option for rendering RoCM and RoCS. No detected flaws or deficiencies have been relevant to the project at this point, though we will be constrained by its capabilities for the duration of the project.

2.5 Assumptions and Dependencies

Users are expected to have the most updated versions of Google Chrome, or Mozilla Firefox. For the reasons stated in the General Constraints section, the chosen browser must have JavaScript enabled. An Internet connection is obligatory.

3 Specific Requirements

This section will explicitly lay out the purposed system design and it's individual requirements.

3.1 External Interface Requirements

3.1.1 User Interfaces

A web user interface will be developed to easily use the tools RoCM provides. A clean and clutter free UI will be necessary due to the complex nature of the physics involved (as to not overwhelm the user). If too much is incorporated as a part of the UI, the user will lose focus on the purpose of RoCM.

3.1.2 Hardware Interfaces

Users will need a device capable of supporting a modern browser and an Internet connection. Experience may depend on hardware capabilities and Internet speeds. No other external hardware dependencies are necessary.

3.1.3 Software Interfaces

RoCM will rely on the open source data visualization library D3.js (v3). A major software interfacing dependency will be JQuery 1.10.2, which will speed up the process of interfacing JavaScript with the HTML DOM elements. A LaTeX equation library, MathJax.js 2.3.1, will be used to convert raw LaTeX code into a formatted mathematical equation. An integration JavaScript library will be used for some of the galactic models. Other mathematical libraries will be added as needed.

3.1.4 Communications Interfaces

RoCM and RoCS will query data from the SOCM database. A bridge between the tools and the database will be developed to smoothly interface the front-end UI with the back-end data. A standardized format will be defined by SOCM.

3.2 Functional Requirements

3.2.1 Rotational Curve Simulator (RoCM)

3.2.1.1 Introduction

RoCM will serve as a visual modeler for our collection of data and contemporary theories for the solution to the rotation curve problem. With observable data as the input (via SOCM), any galaxy can be imported into the tool. The tool will plot obervational data and multiple galactic models together as a graph, as well as enable users to import their own galatic models to test against existing theories. Sliders will allow users to control parameters within their models with realtime visual feedback in the generated graph.

3.2.1.2 Inputs

RoCM will recieve it's galactic data from the SOCM API endpoints. For use of the sliders, users must click and drag JavaScript-generated sliders. The 'LaTeX equation' section will require a properly-formed LaTeX equation (as dictated by the LaTeX standard) if the user desires to use the feature. The 'JavaScript Model Input' section will require a properly-formed JavaScript function in the form v(R). Interaction with SOCM and RoCS will require mouse or cursor input.

3.2.1.3 Processing

Galactic data will be processed by the D3 JavaScript library, which generates and manipulates elements of the HMTL DOM to produce the graphs, sliders, and tables of RoCM. When the user changes the value of the sliders, each model that uses that parameter will be recalculated on the fly.

3.2.1.4 Outputs

RoCM produces a graph of Rotational Velocity (in km/s) over Galactocentric Distance (in kpc) with sliders to dynamically manipulate the graph.

3.2.1.5 Error Handling

If RoCM is unable to connect to SOCM, the page will detect the problem and issue an error message to the user. Errors resulting from JavaScript being disabled will be handled at a higher level (in the HMTL pages).

3.2.2 rotationcurve.org Website

3.2.2.1 Introduction

Currently, scholars must sift through peer-reviewed articles and gather galaxy data one-by-one. The rotationcurve.org website will help generalize the work being done on galaxy research and provide one central repository for researchers.

3.2.2.2 Inputs

Outside of the constituent modules of RoCMSOCM, the website will only require a user's mouse or cursor input for navigation. The website will be comprised of HTML, JavaScript, and CSS files.

3.2.2.3 Processing

The web browser will process all necessary web requests and JavaScript will create the necessary navigational buttons on demand.

3.2.2.4 Outputs

The resulting product will be a full graphical interface for the RoCM, SOCM, and RoCS modules. Users will be able to interact with the graphs and models in realtime, aiding in their understanding and exploration of the topic.

3.2.2.5 Error Handling

While the Internet browser will handle any issues regarding connection to the website, the website itself will detect and report on any errors regarding JavaScript. A message will appear for users who have JavaScript disabled which will prompt them to enable it for the best experience. Any errors with HTTP will be handled by the web server hosting rotationcurve.org.

3.3 Classes/Objects

1. RoCM.html
   (a) Curve Plot div
   (b) Parameter Table div
   (c) Parameter Slider div
   (d) LaTeX Equation div
   (e) JavaScript Model Input div
2. RoCM.js
   (a) CurvePlot.js
   (b) ParamTable.js
   (c) ParamSlider.js
   (d) LaTeXEquation.js
   (e) JSModelInput.js
   (f) GalacticModel.js
   (g) Params.js
   (h) Constants.js
   (i) Conversion.js
3. RoCMWebApplication Ruby on Rails Application
   (a) RoCMModel.rb
   (b) RoCMView.rb
   (c) RoCMController.rb

3.4 Non-Functional Requirements

3.4.1 Performance

RoCM should be able to update each model within the curve plot with less than one second of latency. This is to make sure the flow of the tool isn't compromised because of a slow system.

3.4.2 Reliability

Each widely understood galactic model will come standard with RoCM, providing the users with a set of existing models to test against observable data. The user can rely on the models to be fully functional and validated.

3.4.3 Availability

The website and database will be available 24/7, as long as the hosting machine is online. If the machine crashes, the problem will be debugged and the website will be up as soon as possible.

3.4.4 Security

Security is a low priority for RoCM, but being a web application it is necessary to cover our grounds. If the user imports their own galactic models, RoCM will only use that locally. A injection free system will be developed to minimize security risks.

3.4.5 Maintainability

The modular nature of the design will help developers maintain each encapsulated portion of RoCM. Each module can be updated separately, without intruding on it's neighboring modules. Only if there is a redefinition of inputs and outputs will each module need to be updated accordingly.

3.4.6 Portability

This software can be used on any computer with the latest version of Google Chrome or Mozilla Firefox. Even Android smart phones will be able to run RoCM as a web application. This makes it easy for anyone, anywhere to use RoCM.

3.5 Inverse Requirements

RoCM stresses that the user shouldn't need to learn any proprietary material in order to use the system. Although, an understanding of rotation curves and their implications in the astrophysics community would be highly recommended to gain the full functionality out of RoCM.

3.6 Design Constraints

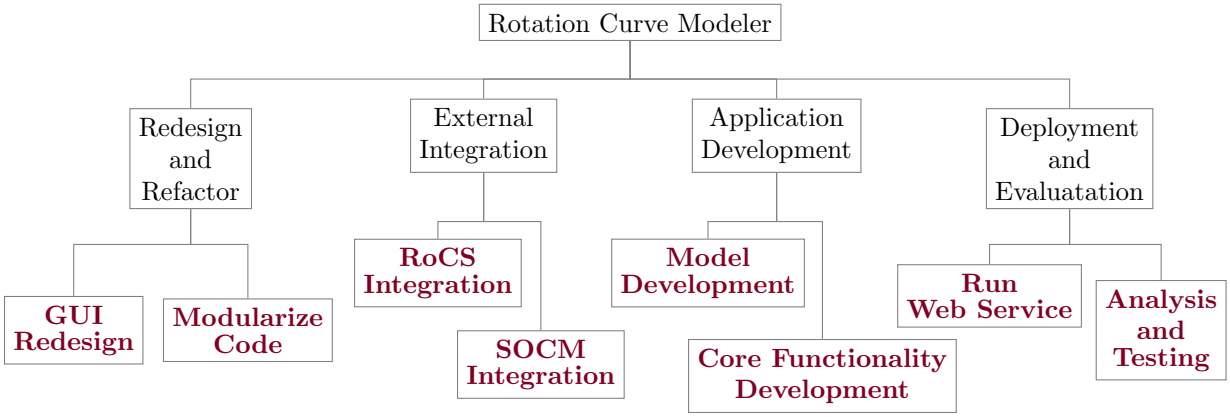
The constrains will be directed at the internet browser that the user chooses to run RoCM. Because this software has multiple features, each software module will be encapsulated in a way to be reused elsewhere.

3.7 Logical Database Requirements

SOCM will be the database portion of RoCM. *See the SOCM Software Requirements Specification.*

4 Project Planning and Risk Management

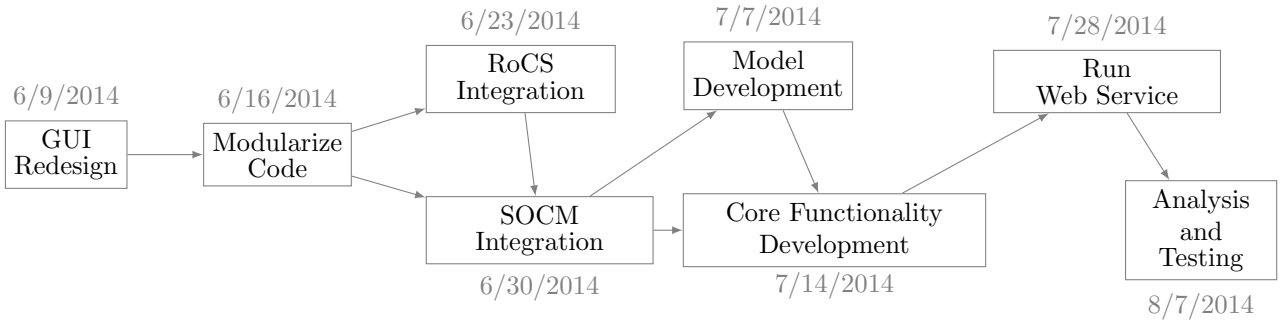
4.1 Work Breakdown Structure



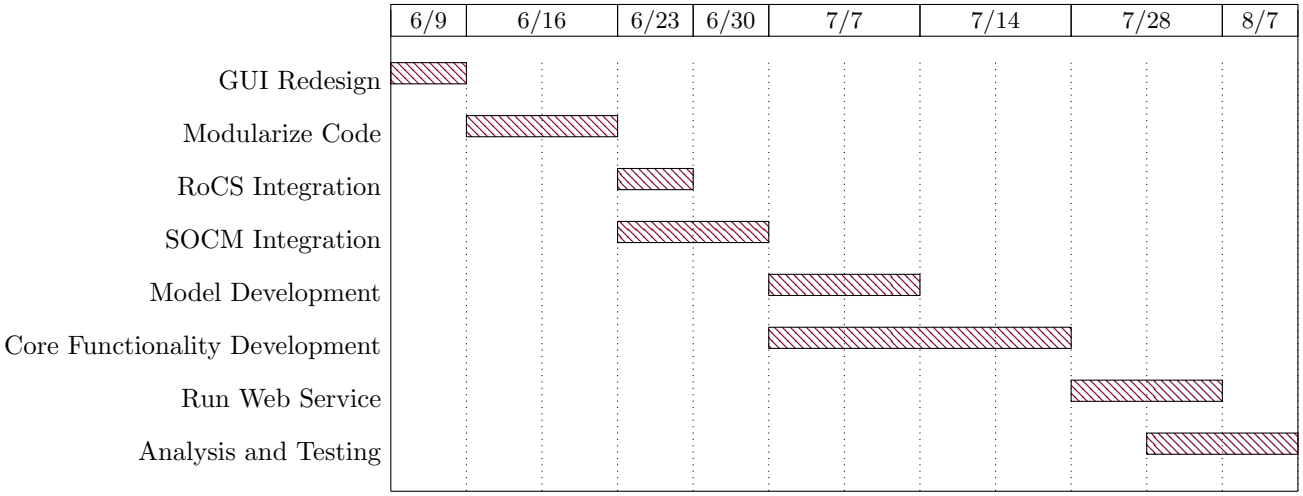
4.2 Time Estimation

Task	Number of Weeks
GUI Redesign	1.5
Modularize Code	1
RoCS Integration	1
SOCM Integration	2
Model Development	1
Core Functionality Development	2
Run Web Service	1
Analysis and Testing	1
Total	**10.5**

4.3 Activity Sequencing Diagram



4.4 Gantt Chart



4.5 Risk Management

Risk	Likelihood	Consequences	Total	Fix
Website crash	1	9	**9**	The team will be responsible for restarting the server.
Slow browser	2	2	**4**	The user can move to a faster computer.

A Appendix

A.1 Design Concept

A tentative conceptual design of the system is as followed: