

15/15

# SYSTEMS AND SOFTWARE REQUIREMENTS SPECIFICATION (SSRS)

Version A.2, September 2010

## FOREWORD

This document was written to provide software development projects with a template for generating a System and Software Requirements Specification (SSRS). This document is based on a template originally written by the U.S. Navy Research, Development, Test and Evaluation Division in June 1997 in accordance with the MIL-STD-498 DID (DI-IPSC-81433). The template was updated by the University of Idaho's Center for Secure and Dependable Systems (CSDS) in June 2008 to adhere to IEEE Std. 830-1998, *IEEE Recommended Practice for Software Requirements Specifications*<sup>1</sup>, and IEEE Std. 12207-2008, *Systems and Software Engineering – Software Life Cycle Processes*<sup>2</sup>. It was then adapted in September 2008 for use in UI CS 383.

The SSRS template begins on the next page. Just throw away this page and enter your project specifications into the following template. Don't forget to change the headers and footers as necessary.

## DOCUMENT CONVENTIONS

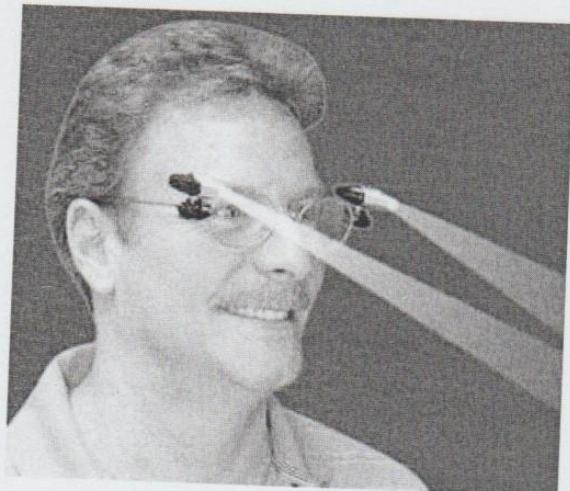
| Text | Replace this text with your project specification text.

*text in italics* Notes or instructions to the author. Delete in final format.

D PT

<sup>1</sup> IEEE Std. 830-1998, *Recommended Practice for Software Requirements Specifications*, Institute of Electrical and Electronics Engineers, 345 East 47<sup>th</sup> St. New York, NY, USA, 10017-2394.

<sup>2</sup>ISO/IEC 12207, IEEE Std. 12207-2008, *Systems and software engineering – Software life cycle processes*, 2<sup>nd</sup> ed., Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ, USA, 08854.



SYSTEMS AND SOFTWARE REQUIREMENTS SPECIFICATION (SSRS) FOR  
Goofy Lights Editor

Version 0.1  
March 7th, 2017

Prepared for:  
[insert company name, address, and contact info]

Prepared by:  
Nicholas Blackburn  
Joseph Chereck  
Michael Madsen  
Ruth Park  
Jocelyn Stadler  
Lance Werhane  
University of Idaho  
Moscow, ID 83844-1010

Team 4

Googy Lights Editor SSRS

## RECORD OF CHANGES

\*A - ADDED M - MODIFIED D - DELETED

**GOOFY LIGHTS EDITOR SSRS  
TABLE OF CONTENTS**

Section	Page
<b>1</b> Introduction	<b>1</b>
1.1 IDENTIFICATION . . . . .	1
1.2 PURPOSE . . . . .	1
1.3 SCOPE . . . . .	1
1.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS . . . . .	1
1.5 REFERENCES . . . . .	2
1.6 OVERVIEW AND RESTRICTIONS . . . . .	2
<b>2</b> OVERALL DESCRIPTION	<b>3</b>
2.1 PRODUCT PERSPECTIVE . . . . .	3
2.2 PRODUCT FUNCTIONS . . . . .	3
2.3 USER CHARACTERISTICS . . . . .	3
2.4 CONSTRAINTS . . . . .	3
2.5 ASSUMPTIONS AND DEPENDENCIES . . . . .	4
2.6 SYSTEM LEVEL (NON-FUNCTIONAL) REQUIREMENTS . . . . .	4
2.6.1 Site dependencies . . . . .	4
2.6.2 Safety, security and privacy requirements . . . . .	4
2.6.3 Performance requirements . . . . .	4
2.6.4 System and software quality . . . . .	5
2.6.5 Packaging and delivery requirements . . . . .	5
2.6.6 Personnel-related requirements . . . . .	5
2.6.7 Training-related requirements . . . . .	5
2.6.8 Logistics-related requirements . . . . .	5
2.6.9 . . . . .	6
2.6.10 Other requirements . . . . .	6
2.6.11 Precedence and criticality of requirements . . . . .	6
<b>3</b> SPECIFIC REQUIREMENTS	<b>7</b>
3.1 EXTERNAL INTERFACE REQUIREMENTS . . . . .	7
3.1.1 Hardware Interfaces . . . . .	7
3.1.2 Software Interfaces . . . . .	7
3.1.3 User Interfaces . . . . .	7
3.2 SYSTEM FEATURES . . . . .	1
3.2.1 Use Case Diagrams . . . . .	1
3.2.2 System feature 1: Animation Editor . . . . .	4
3.2.3 System feature 2: Frames Viewer . . . . .	5
3.2.4 System feature 3: Menu Bar . . . . .	5
<b>4</b> PROJECTED DEVELOPMENT TIMELINE	<b>1</b>
<b>5</b> REQUIREMENTS TRACEABILITY	<b>1</b>
<b>6</b> APPENDIX A.	<b>1</b>

numbers



# 1 Introduction

The Goofy Lights Editor was inspired by the Tower Lights Editor, created for the University of Idaho to use on Theopolis Tower to create lightshows. Through this inspiration, a new project was concieved; controlling LED lights mounted to glasses worn by the University of Idaho's marching band. The goal of this project is to allow for easy creation of light shows on the marching band's "Goofy Glasses".

## 1.1 IDENTIFICATION

The software system being considered for development is referred to as "Goofy Lights Editor". The customer providing specifications for the system is Dr. Rinker of the University of Idaho Computer Science department. The ultimate customer, or end-user, of the system will be the University of Idaho marching band, or other teams creating shows with the marching band. This is a new project effort, so the version under development is version 0.1.

## 1.2 PURPOSE

The purpose of the system under development is to allow for easy creation of a choreographed light show utilizing the Goofy Glasses, in the form of an output file which can be understood by Goofy Glasses Software. While the system will be used by the University of Idaho marching band and affiliated peoples, this document is intended to be read and understood by UICS software designers and coders. The document will also be approved by Mr. Bruce Bolden of University of Idaho's Computer Science Department.

## 1.3 SCOPE

This project is a new creation inspired by the Tower Animation Software created for Robert Rinker of the University of Idaho Computer Science Department. Developed by Team 4 of Mr. Bruce Bolden's Computer Science 383 class. There is a readme file included with the program explaining basic use.

## 1.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

<i>Term or Acronym</i>	<i>Definition</i>
<i>Alpha test</i>	<i>Limited release(s) to selected, outside testers</i>
<i>Beta test</i>	<i>Limited release(s) to cooperating customers wanting early access to developing systems</i>
<i>Final test</i>	<i>aka, Acceptance test, release of full functionality to customer for approval</i>
<i>DFD</i>	<i>Data Flow Diagram</i>
<i>SDD</i>	<i>Software Design Document, aka SDS, Software Design Specification</i>
<i>SRS</i>	<i>Software Requirements Specification</i>
<i>SSRS</i>	<i>System and Software Requirements Specification</i>
<i>TAN</i>	<i>Tower Animation filetype</i>

## **1.5 REFERENCES**

*This section shall list full bibliographic citations of all documents referenced in this report. This section shall also identify the source for all materials not available in printed form (e.g., web-based information) and list the complete URL along with owner, author, posting date, and date last visited.*

*[ To be expanded when more information is available]*

## **1.6 OVERVIEW AND RESTRICTIONS**

*This document is intended for the University of Idaho Tower of Lights only. For use outside of this purpose please check with the CS Department at the University of Idaho for rights.*

*This document is for limited release only to UI CS personnel working on the project and Mr. Bruce Bolden.*

*Section 2 of this document describes the system under development from a holistic point of view. Functions, characteristics, constraints, assumptions, dependencies, and overall requirements are defined from the system-level perspective.*

*Section 3 of this document describes the specific requirements of the system being developed. Interfaces, features, and specific requirements are enumerated and described to a degree sufficient for a knowledgeable designer or coder to begin crafting an architectural solution to the proposed system.*

*Section 4 provides the requirements traceability information for the project. Each feature of the system is indexed by the SSRS requirement number and linked to its SDD and test references.*

*Sections 5 and up are appendices including original information and communications used to create this document.*

## 2 OVERALL DESCRIPTION

This section of the document should describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides the background for those requirements, which are defined in detail in Section 3.

### 2.1 PRODUCT PERSPECTIVE

This subsection of the document should put the product into perspective with other related products. If the product is independent and totally self-contained, it should be so stated here. If the document describes a product that is a component of a larger system, then this subsection should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software. A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful.

The perspective of the project is related to the Tower lights software, its primary function is to edit/animate sequences of lights. This utilizes the wireless aspect of the goofy glasses which is out of the project scope. This software is to provide an editor for the lights, which will convert to a file and later be translated to the glasses.

### 2.2 PRODUCT FUNCTIONS

This subsection of the document should provide a summary of the major functions that the software will perform. For the sake of clarity The functions should be organized in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time. Textual or graphical methods can be used to show the different functions and their relationships. Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables.

The function of the Goofy Glasses Editor is similar to the tower lights. However, this project adds new challenges and functionality that is created by having dynamic size formation. Since the lights are full range RGB LEDs the functionality of changing color is a great opportunity to add. The end goal is for a band to enable presets that will project letters in uppercase. This can help the user expedite their own projects. Another way this will be done is to add the ability to shift the whole grid in different directions without individually moving the cells. One of the last functions is to have the ability to save a frame or sequence, not the whole project. This will allow the user to make something and use it on other projects. Lastly all the necessary functions like, saving, adjusting timing, creating new files, and warnings.

### 2.3 USER CHARACTERISTICS

This subsection of the document should describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise. It should not be used to state specific requirements, but rather should provide the reasons why certain specific requirements are later specified in Section 3 of this document.

The customers/users for the Goofy Lights Editor is the staff and students at the University of Idaho. This editor will allow the customer to make specific sequences of lights. There is no training required to use the Goofy Lights Editor. There will be cautionary warnings to keep the users from deleting their progress without saving, doing something that is out the bounds, or something out of the scope of the software.

### 2.4 CONSTRAINTS

This subsection of the document should provide a general description of any other items that will limit the developer's options. These include: a) Regulatory policies; b) Hardware limitations (e.g., signal timing

*requirements); c) Interfaces to other applications; d) Parallel operation; e) Audit functions; f) Control functions; g) Higher-order language requirements; h) Signal handshake protocols; i) Reliability requirements; j) Criticality of the application; k) Safety and security considerations.*

*Time and lack of experience seem to be the few restrictions that the design team are experiencing currently. The hardware is out of the project scope and the customer has experience using a similar structure with the tower lights.*

## **2.5 ASSUMPTIONS AND DEPENDENCIES**

*This subsection of the document should list each of the factors that affect the requirements stated in the document. These factors are not design constraints on the system and/or software but are, rather, any changes to them that can affect the requirements in the document. For example, an assumption may be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the document would then have to change accordingly.*

*/ insert your text here /*

## **2.6 SYSTEM LEVEL (NON-FUNCTIONAL) REQUIREMENTS**

*This subsection of the document should identify system level (whole, not functional) requirements that impact the construction, operation, packaging and delivery of the system and software.*

### **2.6.1 Site dependencies**

*This paragraph shall specify site-dependent operational parameters and needs (such as parameters indicating operation-dependent targeting constants or data recording). . The requirements shall include, as applicable, number of each type of equipment, type, size, capacity, and other required characteristics of processors, memory, input/output devices, auxiliary storage, communications/ network equipment, and other required equipment or software that must be used by, or incorporated into, the system. Examples include operating systems, database management systems, communications/ network software, utility software, input and equipment simulators, test software, and manufacturing software. The correct nomenclature, version, and documentation references of each such device or software item shall be provided.*

*Since the Hardware portion of the Goofy Lights is not to be designed by the software design team, nor any instructions provided about the interface of the glasses. Therefor, the hardware dependencies are not within the project scope. The design team has the equipment that is used to program and interface with GitHub.*

### **2.6.2 Safety, security and privacy requirements**

*This paragraph shall specify the system requirements, if any, concerned with maintaining safety, security and privacy. These requirements shall include, as applicable, the safety, security and privacy environment in which the system must operate, the type and degree of security or privacy to be provided, and the criteria that must be met for safety/security/privacy certification and/or accreditation.*

*The Editor does not need security precautions to operate, and safety of the user is not a concern because no hazards are included in the project.*

### **2.6.3 Performance requirements**

*This paragraph should specify both the static and the dynamic numerical performance requirements placed on the soft ware or on human interaction as a whole. Static numerical requirements may include the following:  
a) The number of terminals to be supported; b) The number of simultaneous users to be supported; c) Amount and type of information to be handled. Dynamic numerical requirements may include, for example, the numbers of transactions and tasks and the amount of data to be processed within certain time periods for*

*both normal and peak workload conditions. All of these requirements should be stated in measurable terms. For example, "95% of the transactions shall be processed in less than 1msec."*

*The performance of the Editor is it should run stable for (max number of cells), for a (max length of sequence). It should be able to store as many (stamps) as is needed. It should be able to change every cell for every frame.*

#### **2.6.4 System and software quality**

*This paragraph shall specify the requirements, if any, concerned with hardware and software quality factors identified in the contract. Examples include quantitative requirements regarding the system's functionality (the ability to perform all required functions), reliability (the ability to perform with correct, consistent results), maintainability (the ability to be easily corrected), availability (the ability to be accessed and operated when needed), flexibility (the ability to be easily adapted to changing requirements), portability (the ability to be easily modified for a new environment), reusability (the ability to be used in multiple applications), testability (the ability to be easily and thoroughly tested), usability (the ability to be easily learned and used), and other attributes.*

*This shall be a stable piece of software that runs smoothly with little to no interruptions.*

#### **2.6.5 Packaging and delivery requirements**

*This paragraph shall specify the requirements, if any, for packaging, labeling, handling and delivery of the system being developed to the customer.*

*The executable system and all associated documentation (i.e., SSRS, SDD, code listing, test plan (data and results), and user manual) will be delivered to the customer on CD's and/or via email, as specified by the customer at time of delivery. Although document "drops" will occur throughout the system development process, the final, edited version of the above documents will accompany the final, accepted version of the executable system.*

#### **2.6.6 Personnel-related requirements**

*This paragraph shall specify the system requirements, if any, included to accommodate the number, skill levels, duty cycles, training needs, or other information about the personnel who will use or support the system under development. These requirements shall include, as applicable, considerations for the capabilities and limitations of humans; foreseeable human errors under both normal and extreme conditions; and specific areas where the effects of human error would be particularly serious. Examples include requirements for color and duration of error messages, physical placement of critical indicators or keys, and use of auditory signals.*

*The system under development has no special personnel-related characteristics. Error messages will come about when they come up in development.*

#### **2.6.7 Training-related requirements**

*This paragraph shall specify the system requirements, if any, pertaining to training. Examples include training software, tutorials, or help information to be included in the system.*

*No training materials or expectations are tied to this project other than the limited help screens built into the software and the accompanying user manual.*

#### **2.6.8 Logistics-related requirements**

*This paragraph shall specify the system requirements, if any, concerned with logistics considerations. These considerations may include: system maintenance, software support, system transportation modes, supply-system requirements, impact on existing facilities, and impact on existing equipment.*

*These issues are out of the scope of our project.*

#### **2.6.9**

#### **2.6.10 Other requirements**

*This paragraph shall specify additional system level requirements, if any, not covered in the previous paragraphs.*

*[ insert your text here ]*

#### **2.6.11 Precedence and criticality of requirements**

*This paragraph shall specify, if applicable, the order of precedence, criticality, or assigned weights indicating the relative importance of the requirements in this specification. Examples include identifying those requirements deemed critical to safety, to security, or to privacy for purposes of singling them out for special treatment. If all requirements have equal weight, this paragraph shall so state.*

*[ insert your text here ]*

/

### **3 SPECIFIC REQUIREMENTS**

#### **3.1 EXTERNAL INTERFACE REQUIREMENTS**

##### **3.1.1 Hardware Interfaces**

*To be expanded when more information is provided.*

##### **3.1.2 Software Interfaces**

*To be expanded when more information is provided.*

##### **3.1.3 User Interfaces**

*When the user opens the graphical interface they will be brought to the Goofy Lights Editor screen. The user must have the ability to customize frames by being able to interact with a grid structure that allows the changing of colors of individual boxes of the grid.*

*The user will have the ability to create a new project, save a project, and open an existing project.*

*When editing a project the user will have the ability to insert and delete individual frames, change the position of a previously created frame in the animation, set the time interval between frames. The user will be able to scroll through all the frames in the current animation, and view the final animation.*



*External Interface Requirements*

*Hardware Interfaces*

Name	Source/Destination	Description	Type/range	Dependencies

*Software Interfaces*

Name	Source/Destination	Description	Type/range	Dependencies

*User Interfaces*

Name	Source/Destination	Description	Type/range	Dependencies
Goofy Lights Editor		Opening menu screen containing the menu bar, animation editor, and frames viewer	Graphical user interface	
Menu Bar	Goofy Lights Editor	Drop down menus for file, edit, and playback	Menu driven	
Animation Editor	Goofy Lights Editor	Displays current frame and tools for editing and animating the frame	Graphical user interface	
Frames Viewer	Goofy Lights Editor	Displays frames of animation and allows for deletion and insertion of frames.	Graphical user interface	

*Other Communication Interfaces*

Name	Source/Destination	Description	Type/range	Dependencies

/

## 3.2 SYSTEM FEATURES

Functional requirements should define the fundamental actions (i.e., features) that must take place in the software in accepting and processing the inputs and in processing and generating the outputs. These requirements are given in the form of **Use Cases** where possible, denoting a concrete use (discrete user-performable task) of the system. Use case diagrams are followed by use case descriptions, followed by any non-task features. Non-task features are generally listed as "shall" statements starting with "The system shall..." These include: a) Validity checks on the inputs; b) Exact sequence of operations; c) Responses to abnormal situations, including error detection, handling and recovery; d) Parameter specification and usage; e) Relationship of outputs to inputs, including formulas for input to output conversion.

It may be appropriate to partition the functional requirements into sub functions or subprocesses, but that decomposition (here) does not imply that the software design will also be partitioned that way. You should repeat subsections 3.2.i for every specified feature defined for the system or software.

### 3.2.1 Use Case Diagrams

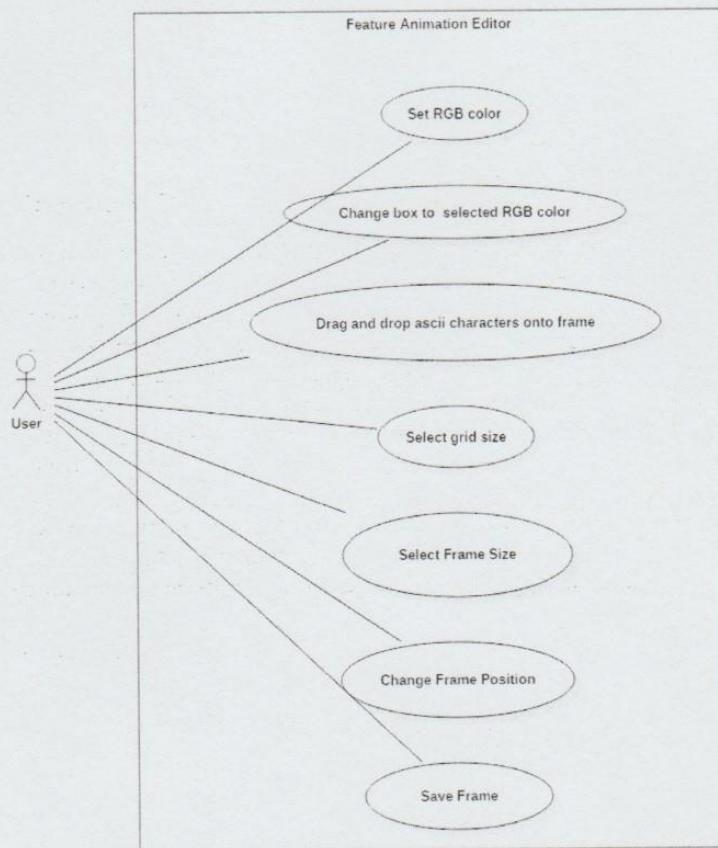


Figure 1: Use case diagram for Animation Editor

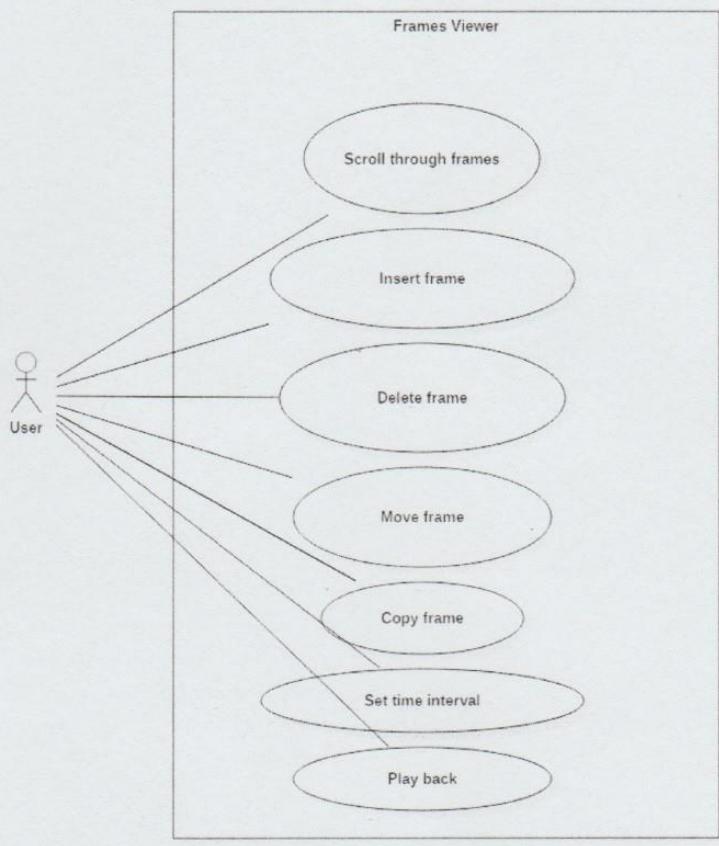


Figure 2: Use case diagram for Frames Viewer

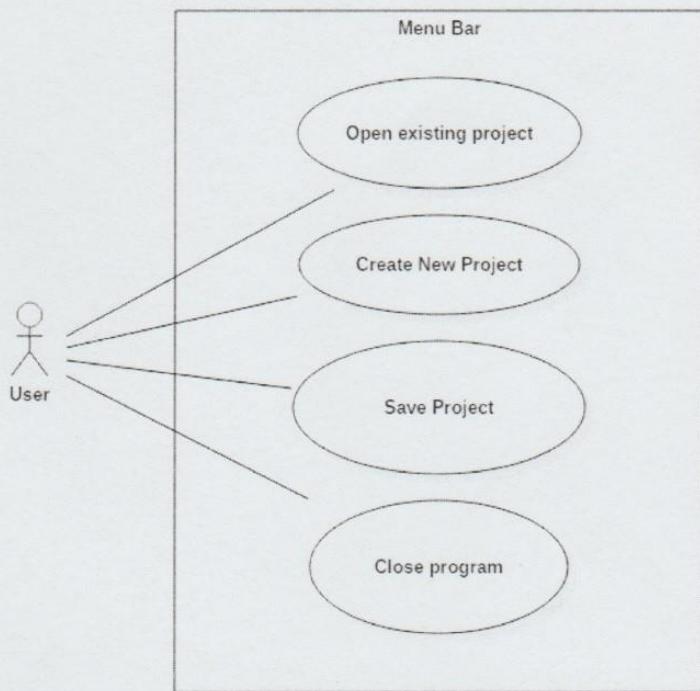


Figure 3: Use case diagram for Menu Bar

### 3.2.2 System feature 1: Animation Editor

Use Case Description	
<b>Name</b>	Animation Editor
<b>Actors</b>	User
<b>Goals</b>	Provide functionality for frame editing before adding to the animation
<b>Preconditions</b>	Execute Goofy Lights Editor program
<b>Summary</b>	Change position size and colors of frame, ability to add preset ASCII characters, and add frame to animation.
<b>Related use cases</b>	Frames Viewer
<b>Steps</b>	<ol style="list-style-type: none"><li>1. Select grid size</li><li>2. Select frame size</li><li>3. Select color</li><li>4. Select boxes</li><li>5. Add to animation</li></ol>
<b>Alternatives</b>	<p>Alternative 1 Precondtion: Previously created frame</p> <ol style="list-style-type: none"><li>1. Move frame</li><li>2. Add to animation</li></ol> <p>Alternative 2 1. Select grid size 2. Select frame size 3. Select color 4. Add ASCII character 5. Add to animation</p>
<b>Postconditions</b>	

?

### 3.2.3 System feature 2: Frames Viewer

<i>Use Case Description</i>
<b>Name</b> Frames Viewer
<b>Actors</b> User
<b>Goals</b> Alter existing animation through frame scrolling and manipulation.
<b>Preconditions</b> Execute Goofy Lights Editor program
<b>Summary</b> Insert, delete, move, and scroll through animation's frames.
<b>Related use cases</b> Animation Editor
<b>Steps</b>
<b>Alternatives</b>
<b>Postconditions</b>

### 3.2.4 System feature 3: Menu Bar

<i>Use Case Description</i>
<b>Name</b> Menu Bar
<b>Actors</b> User
<b>Goals</b> Program manipulation and file organization
<b>Preconditions</b> Execute Goofy Lights Editor program
<b>Summary</b> Options to create, open, and save project or close program
<b>Related use cases</b> None
<b>Steps</b>
1. Select menu item
2. Select option
<b>Alternatives</b>
<b>Postconditions</b> Program performs selected option

## 4 PROJECTED DEVELOPMENT TIMELINE

*The projected development timeline is as follows:*

*March 25th - Program Outline, UI design, and rough division of labor*

*April 2nd - Basic features created but not implemented*

*April 8th - First prototype delivered*

*April 17th - Finished*

## 5 REQUIREMENTS TRACEABILITY

This section shall contain traceability information from each system requirement in this specification to the system (or subsystem, if applicable) requirements it addresses. A tabular form is preferred, but not mandatory.

Feature Name	Req No.	Requirement Description	Priority	SDD	Alpha Release		Beta Release	
					Test Case(s)	Test Res.	Test Case(s)	Test Res.
1	1.1							
	1.2							
	...							
	1.[n]							
	2.1							
	2.2							
	...							
	2.[n]							
	3.1							
	3.2							
2	...							
	3.[n]							
	...							
	[m].1							
	[m].2							
3	...							
	[m.n]							

Priorities are: Mandatory, Low, High

SDD link is version and page number or function name.

Test cases and results are file names and Pass/Fail or % passing.

## 6 APPENDIX A. [insert name here]

*Include copies of specifications, mockups, prototypes, etc. supplied or derived from the customer. Appendices are labeled A, B, ... n. Reference each appendix as appropriate in the text of the document.*

*[ insert appendix A here ]*

**7 APPENDIX B.**

*[ insert appendix B here ]*