

CS3354 Software Engineering
Final Project Deliverable 1

Dejá Vu
Calendar Software

Group Members

Mary Grace Doviak
Ada Zuniga
Riddhi Gunasekaran
Miral Shah
Chase Dannen
Oviya Jayakumar
Luis Castro Ochoa

1. **[5 POINTS]** Please attach here the Final Project draft description (that contains the instructor feedback). It is ok to include a picture of the original document. Address the feedback provided for your proposal by listing what you are did to comply with the proposed changes/requests for additions to your project.
- Although we did not receive specific feedback from the professor, we did talk a lot about our project idea, as well as how we could improve it. Originally we planned on doing a very plain and normal calendar software with just the basic requirements, like different views and adding an event. But we decided that it was critical to add something that made our calendar software stand out. There were many different ideas that we discussed in order to make our software unique, but we ended up choosing to add an image gallery that users can browse and add a photo/gif/emoji to any given event that they have on their calendar. This is similar to the sticker option that snapchat utilizes. We believe that this is a great way to make our calendar unique, as well as more interesting and fun to use.

1. [10 POINTS] Setting up a Github repository:

- 1.1. Each team member should create a GitHub account if you don't already have one.
- 1.2. Create a GitHub repository named 3354-teamName. (whatever your team name will be). DONE
- 1.3. Add all team members, and the two TAs as collaborators. The TAs' GitHub account info are as follows:

TA1 GitHub id: [mukami12](#)

TA1 email: barbara.maweu@utdallas.edu

TA2 GitHub id: [andry5](#)

TA2 email: axk180031@utdallas.edu

- 1.4. Make the first commit to the repository (i.e., a README file with [team name] as its content).
- 1.5. Make another commit including a pdf/txt/doc file named "project_scope". If you choose a predefined topic (one of the 4 topics described in the "Project Topic Ideas" section of this document), the contents of the file should be identical to the corresponding project in this section. If you choose other topics, the contents should follow a similar structure.
- 1.6. Keep all your project related files in your repository as we will check them. Include the URL of your team project repository into your project deliverable 1 report.

Important Note:

- Tasks 1.3 - 1.5 should be performed by different team members. We will check the commit history for these activities.
- Do not include credentials (e.g., UTD ID) in the repository.
- Only commits performed before the deadline will be considered. Do not forget to push your changes after you have done the work!

2. [5 POINTS] Delegation of tasks: Who is doing what

Setting up a Github repository:

1.2 - **Mary Grace Doviak**

1.3 - **Mary Grace Doviak**

1.4 - **Oviya Jayakumar**

1.5 - **Luis Castro Ochoa**

Question 1 - **Mary Grace Doviak**
Question 3 - **Oviya Jayakumar**
Question 4 - **Riddhi Gunasekaran**
Question 5 - **Miral Shah**
Question 6 - **Luis Castro Ochoa**
Question 7 - **Ada Zuniga**
Question 8 - **Chase Dannen**

3. **[5 POINTS]** Which software process model is employed in the project and why.
(Ch 2)

→ A spiral model will be implemented for our calendar software. We wanted a model that would allow for both prototyping and incrementing. As the software is tested by users and given feedback we hope to add to the project to better suit the users. Using the spiral model, the project can be built bit by bit testing part and delivering each part, starting with the most basic part, to the most complex.

4. **[15 POINTS]** Software Requirements including

4.a.) **[5 POINTS]** Functional requirements. To simplify your design, please keep your functional requirements in the range minimum 5 (five) to maximum 7 (seven). (Ch 4)

Functional requirements:

1. Add an event

- 1.1. Check time conflicts
- 1.2. Set start and end times
- 1.3. Set weekly periodical events
- 1.4. Add event categories
 - 1.4.1. Mark color of the event
- 1.5. Add picture
- 1.6. Set event alert

2. **Edit an event**
 - 2.1. Edit event alert
 - 2.2. Edit event categories
 - 2.3. Edit picture
 - 2.4. Edit color for the event
3. **Delete an event**
 - 3.1. Delete event categories
 - 3.2. Delete picture
4. **Change the view**
 - 4.1. Change to Monthly view
 - 4.2. Change to Daily view
 - 4.3. Change to Weekly view

4.b.) [10 POINTS] Non-functional requirements (use **all** non-functional requirement types listed in Figure 4.3 - Ch 4)

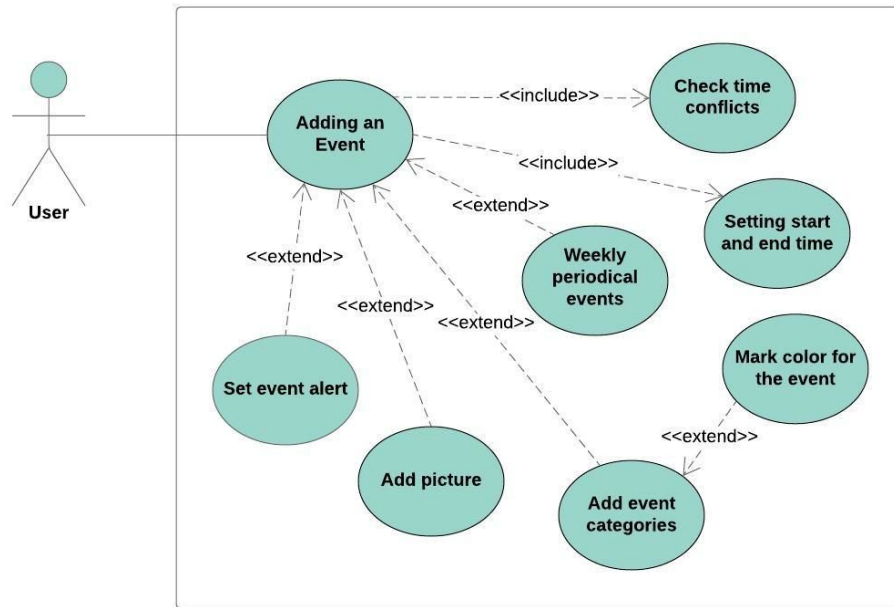
Non - Functional requirements:

1. Product
 - 1.1. Usability
 - The application must be available through web and mobile interfaces.
 - The application will be user-friendly.
 - 1.2. Performance
 - Any action made by the user will be reflected on the application within three seconds of the user's command.
 - 1.3. Efficiency
 - 1.3.1. Space
 - The application can use upto 200 MB memory space.
 - 1.3.2. Dependability
 - The application should be available 24/7.
 - None of the user data will be lost.

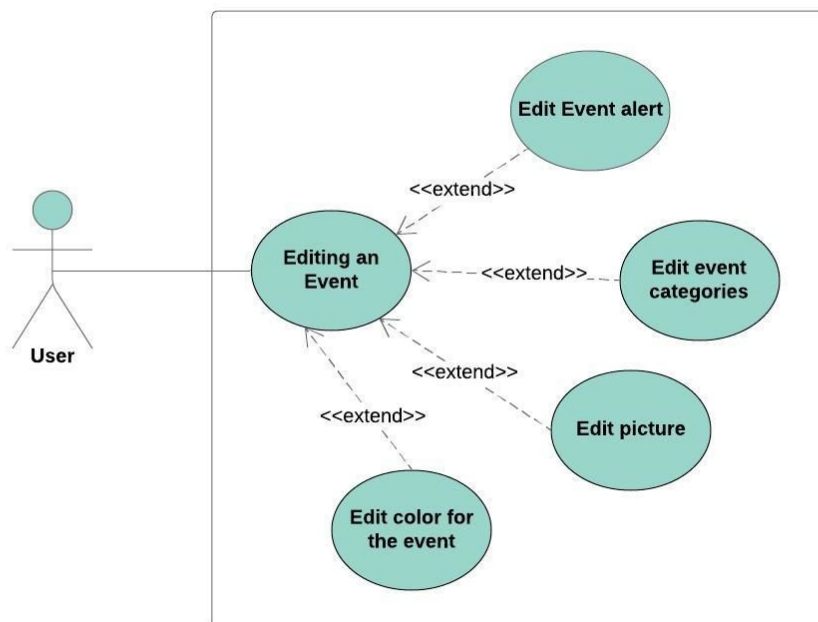
- 1.4. Security
 - All data and pictures stored on the application will be safeguarded and kept confidential.
 - It will also be protected from internal and external threats.
2. Organizational
 - 2.1. Environmental
 - Will run on iOS and Android
 - 2.2. Operational
 - The application will be used as a standard mobile app, with the main form of input being tapping on the screen
 - 2.3. Development
 - Any upgrades or code changes should have similar functionalities and user interface as the older versions.
3. External
 - 3.1. Regulatory
 - The application will comply with all relevant government regulations
 - 3.2. Ethical
 - We will maintain integrity, professional compliance, confidentiality, objectivity and professional behavior.
 - 3.3. Legislative
 - 3.3.1. Accounting
 - The app will be free to download and have extra features the user must pay to use.
 - 3.3.2. Safety/Security
 - Any private information stored on the application will remain confidential for public safety.
 - Unauthorized user access will also be stopped to maintain the safety and integrity of the system.

5. [15 POINTS] Use case diagram – Provide a use case diagram (similar to Figure 5.5) for your project. Please note that there can be more than one use case diagrams as your project might be very comprehensive. (Ch 5 and Ch 7)

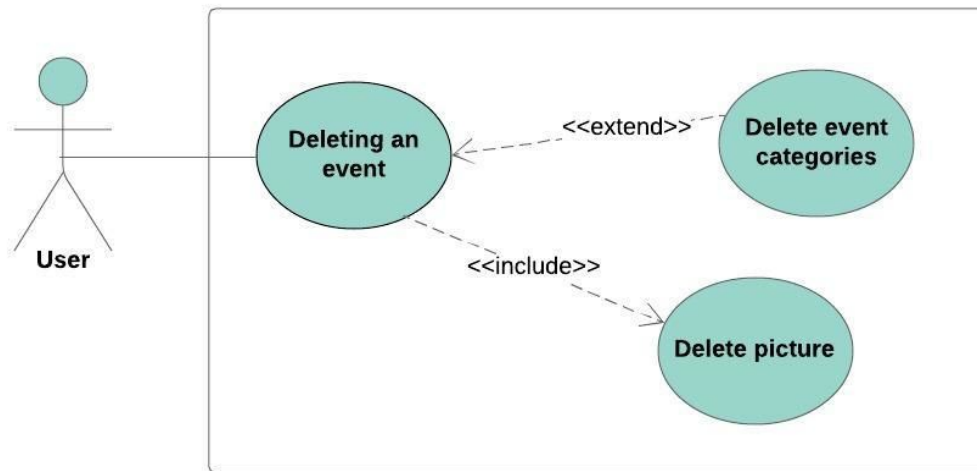
1) Use case for Adding an event:



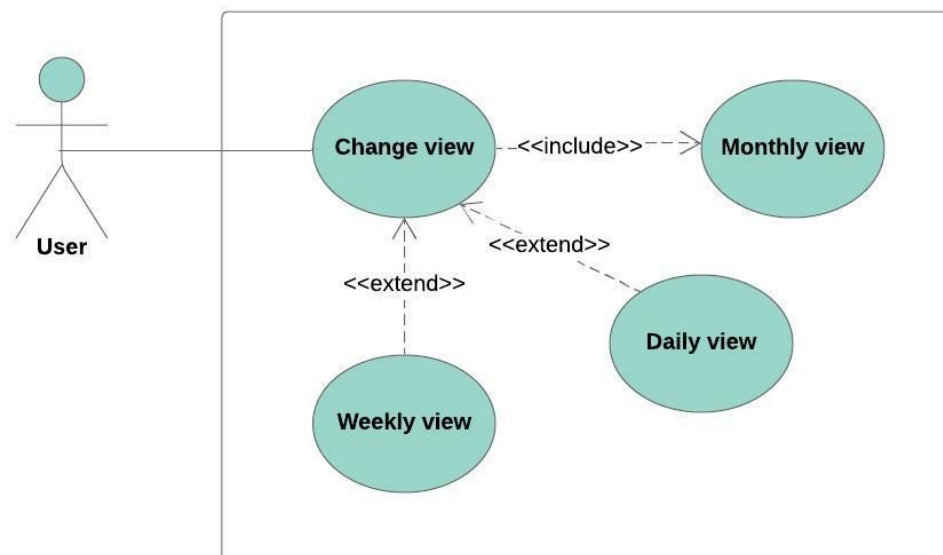
2) Use case for Editing an event:



3) Use case for Deleting an event:

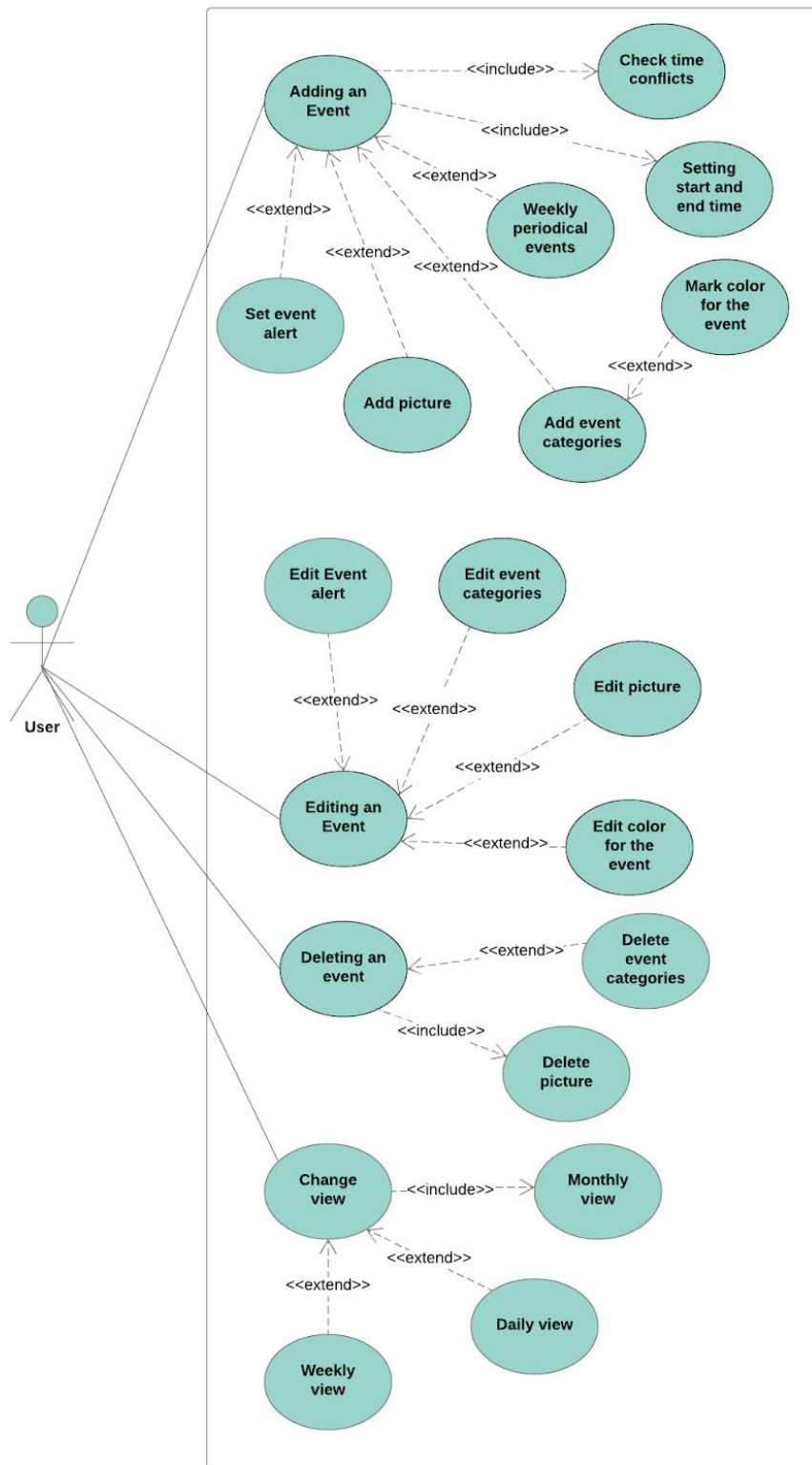


4) Use case for Changing the view:



5) Complete use case diagram:

Dejá Vu - Calendar Application Use Case Diagram

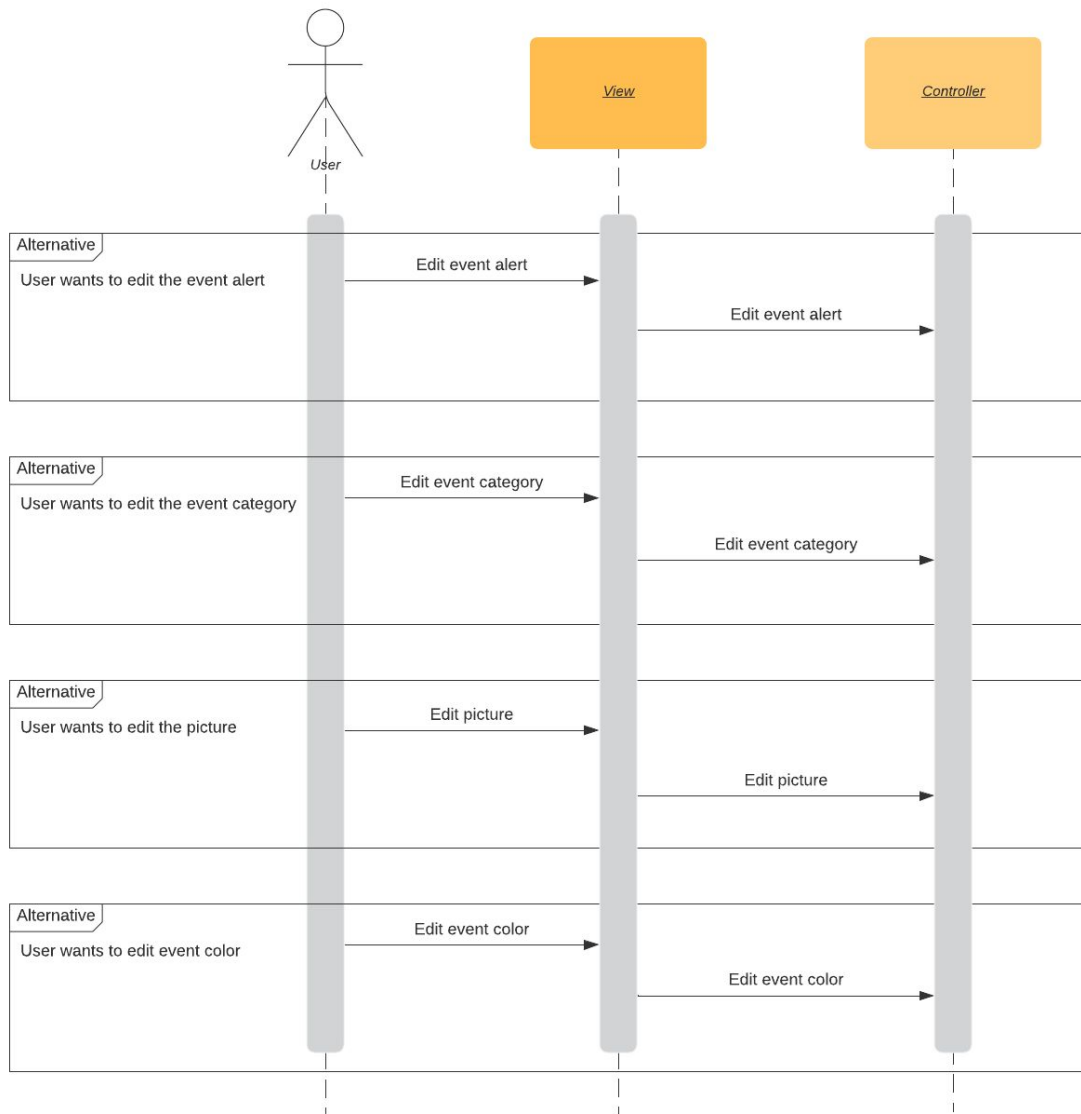


6. [15 POINTS] Sequence diagram – Provide sequence diagrams (similar to Figure 5.6 and Figure 5.7) for each use case of your project. Please note that there **should** be an individual sequence diagram for each use case of your project. (Ch 5 and Ch 7)

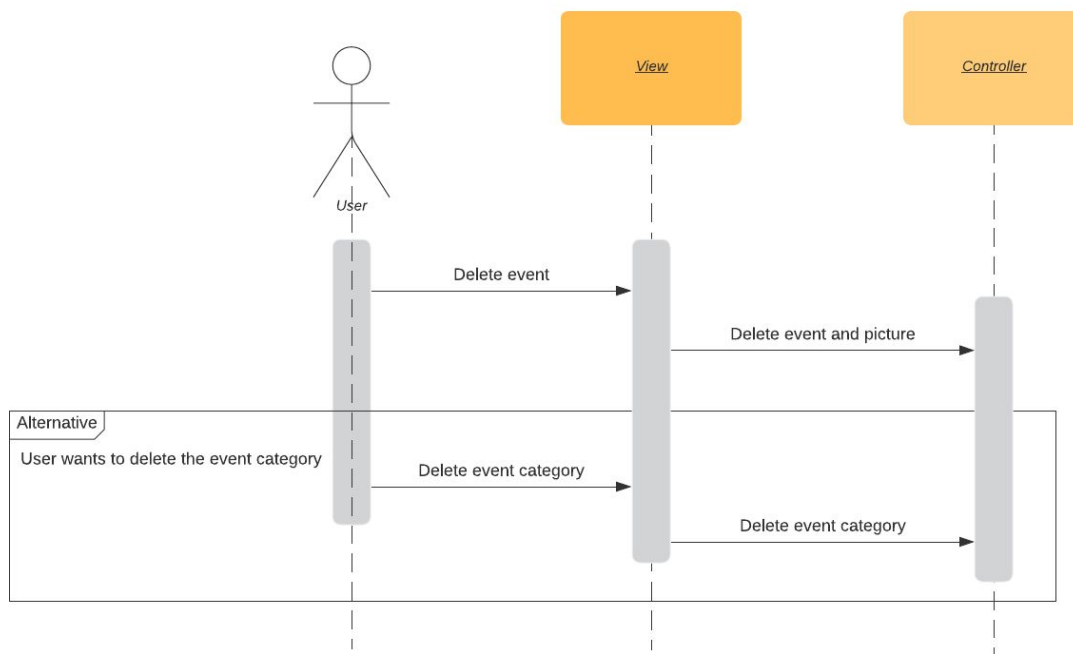
1) Sequence Diagram for Adding an Event



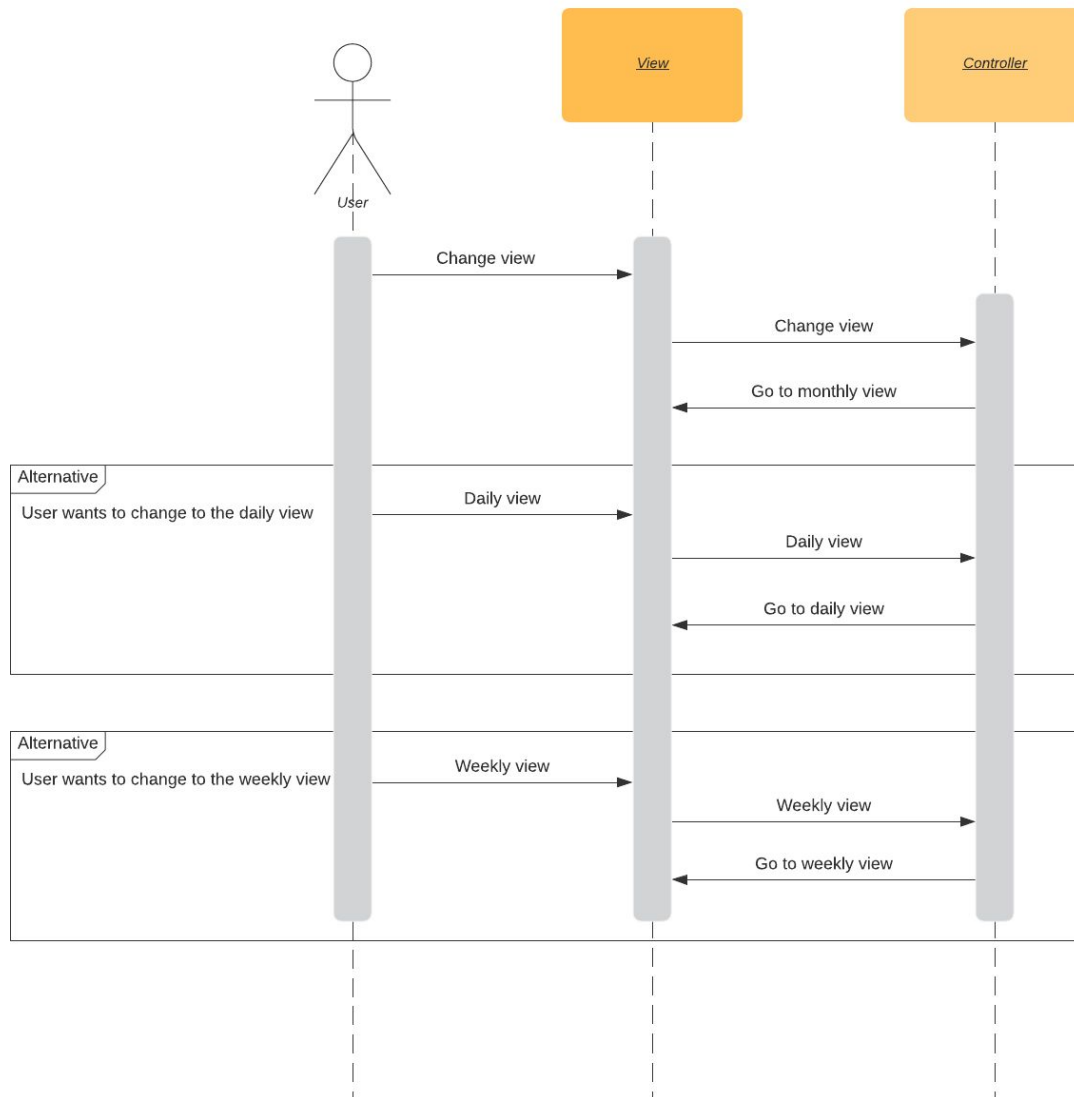
2) Sequence Diagram for Editing an Event



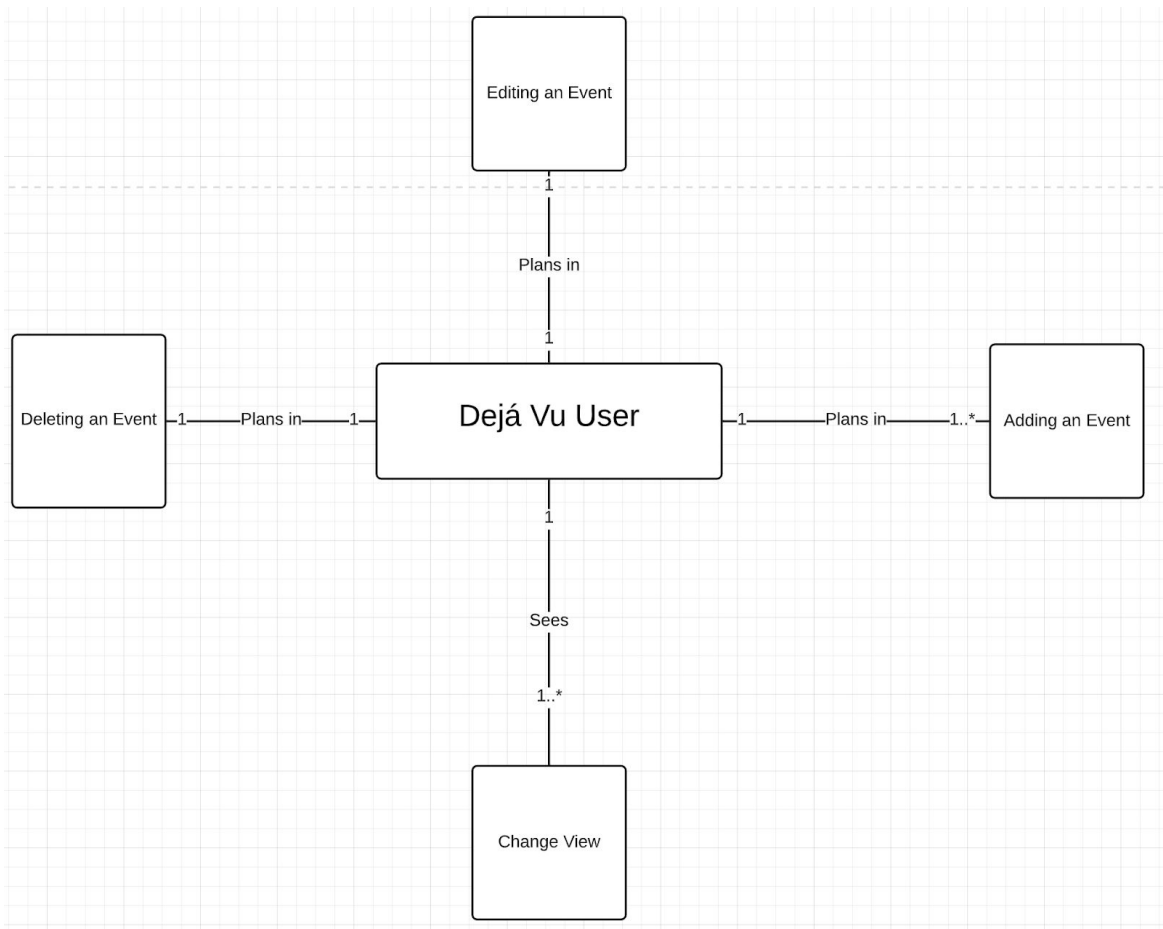
3) Sequence Diagram for Deleting an Event

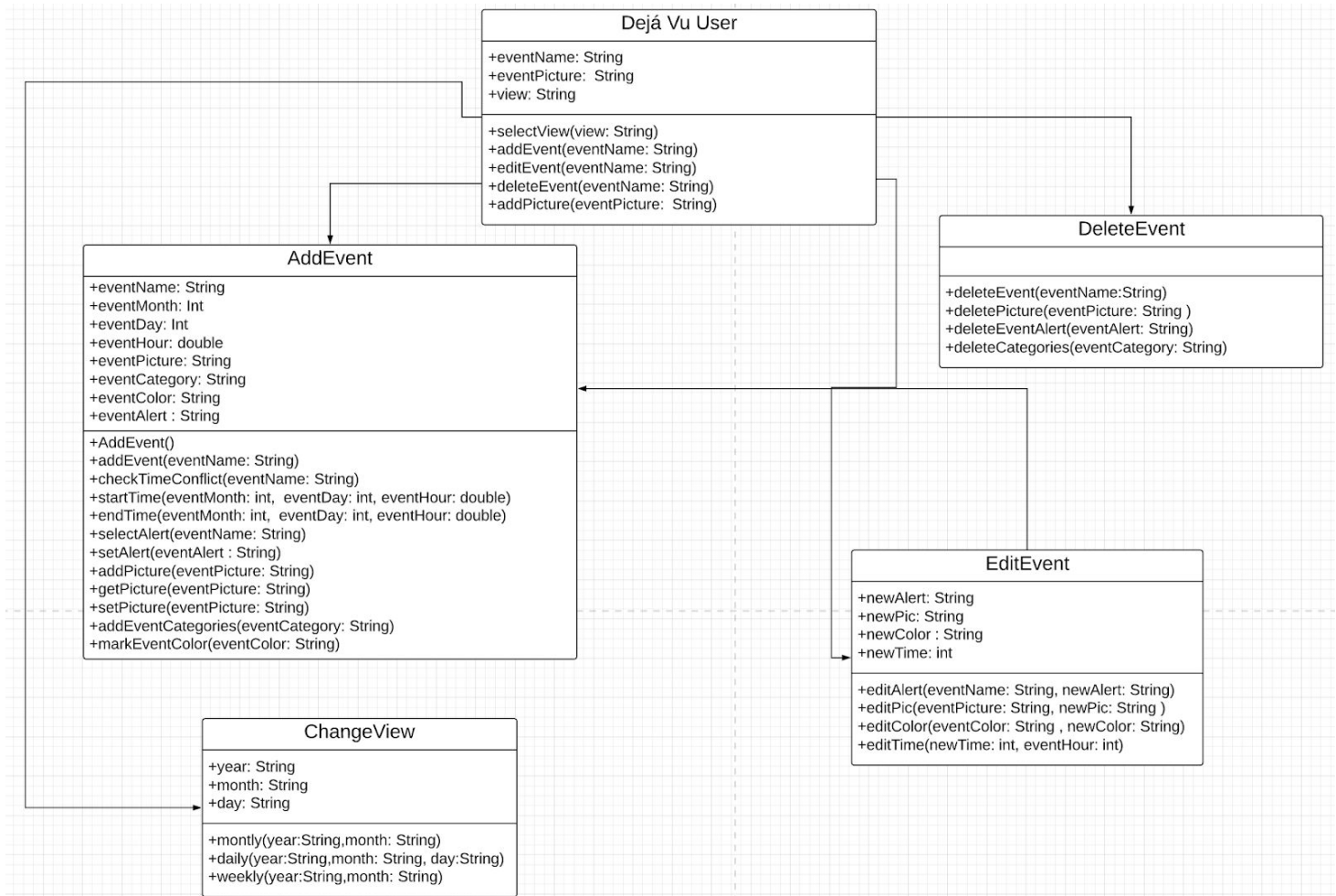


4) Sequence Diagram for Changing the View



7. **[15 POINTS]** Class diagram – Provide a class diagram (similar to Figure 5.9) of your project. The class diagram should be unique (only one) and should include all classes of your project. Please make sure to include cardinalities, and relationship types (such as generalization and aggregation) between classes in your class diagram. Also make sure that each class has class name, attributes, and methods named (Ch 5).

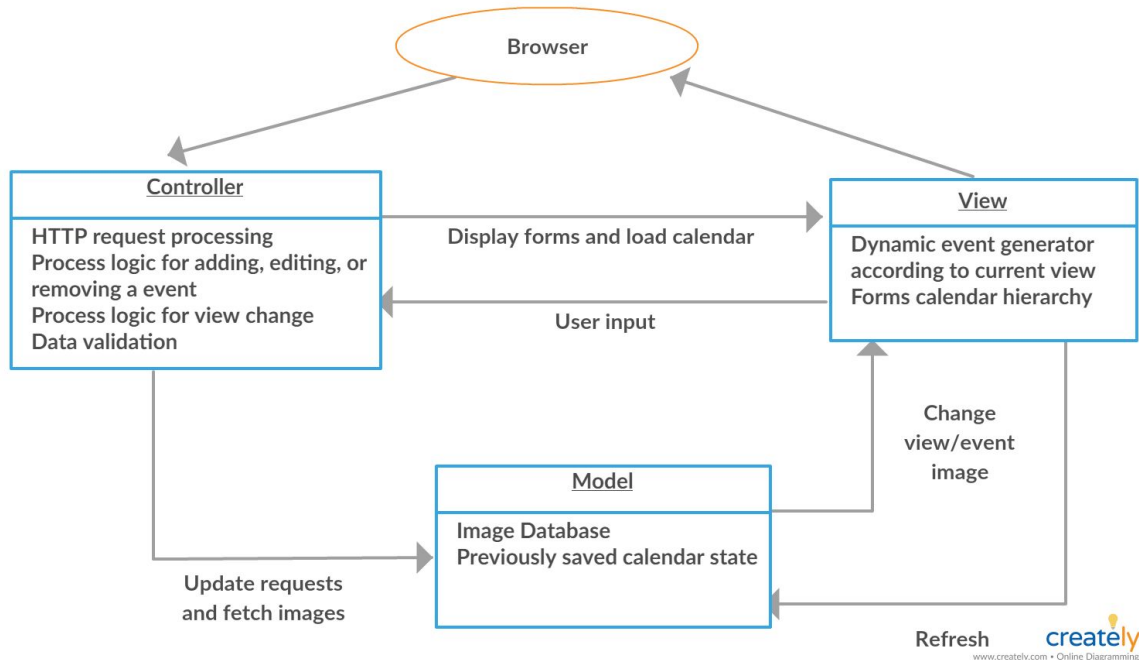




8. [15 POINTS] Architectural design – Provide an architectural design of your project. Based on the characteristics of your project, choose and apply **only one** appropriate architectural pattern from the following list: (Ch 6 section 6.3)
- Model-View-Controller (MVC) pattern (similar to Figure 6.6)
 - Layered architecture pattern (similar to Figure 6.9)
 - Repository architecture pattern (similar to Figure 6.11)
 - Client-server architecture pattern (similar to Figure 6.13)
 - Pipe and filter architecture pattern (similar to Figure 6.15)

Answer:

MVC Architectural Design



(NO EXTENSION IS POSSIBLE ON FINAL PROJECT DELIVERABLE 1 SUBMISSION DUE DATE). No min/max page, font type restrictions.

IMPORTANT NOTE: Use an automated tool for drawing all diagrams required in each deliverable. No manual drawing please.

Submit one copy per group.

Please make sure the whole report is in your own words. Even if you refer to a scholar work, the words should not be exact copy and pastes, but should be rephrased in your own words. Otherwise is called plagiarism and warrants disciplinary action. Please refer to the syllabus about course policy on plagiarism.

Project Topic Ideas:

You may base your project idea on a fictitious institution, or an existing one. If it is an existing institution, please make sure that you comply with company privacy rules and get ALL necessary permissions from company members while obtaining and processing data.

Following is a list of some suggested topics. Some of these topics are described in more detail and others by title only. Feel free to come up with your own project topic ideas, if you want.

1. A Calendar Software

1.1 Views

- 1.1.1 Monthly view: show all days in a month, and event snippet for each day
- 1.1.2 Weekly view: show all days in a week, and event snippet for each day
- 1.1.3 Daily view: show all events in a day, sorted by their starting time
- 1.1.4 Agenda view (optional): show all events in the future as a list

1.2 Events

- 1.2.1 Add an event with starting and ending time
- 1.2.2 Check time conflicts when adding events
- 1.2.3 Add weekly periodical events
- 1.2.4 Edit & delete events
- 1.2.5 Event alert (optional)
- 1.2.6 Add/delete event categories
- 1.2.7 Color marking for different category of events

- 1.3 Share (optional)
 - 1.3.1 Send event to other calendar users (requiring their permission) through internet
 - 1.4 Other
 - 1.4.1 Holidays & weekends should be in special colors
 - 1.4.2 Zoom in/out, and scroll support when necessary
- 2 SMS Messenger Software
- 2.1 Messages
 - 2.1.1 View, edit, and delete messages, save a message under edition as draft
 - 2.1.2 Send & receive messages
 - 2.1.3 Reply & forward messages
 - 2.1.4 Search messages by text query
 - 2.1.5 Send to multiple receivers (optional)
 - 2.1.6 Scheduled message (optional)
 - 2.1.7 Auto reply (optional)
 - 2.2 Message organization
 - 2.2.1 Categorize messages by phone number (contact name)
 - 2.2.2 Conversation view: view all messages between you and a certain contact, sorted by sending/receiving time
 - 2.3 Other
 - 2.3.1 Zoom in/out and scroll whenever necessary
- 3 Contact Manager Software
- 3.1 Contacts
 - 3.1.1 Add, view, edit, delete contacts
 - 3.1.2 Support multiple phone numbers
 - 3.1.3 Support adding a photo label for a contact
 - 3.1.4 Search contacts by contact name
 - 3.1.5 Blacklist (block SMS and Phones)
 - 3.1.6 Directly make phone calls and send SMS from a contact view
 - 3.2 Contacts Organization
 - 3.2.1 Add contact groups
 - 3.2.2 Manage contact groups (add/remove contacts)
 - 3.2.3 Sort contacts by name / group name
 - 3.3 Contacts Storage (optional)
 - 3.3.1 Export contacts to file
 - 3.3.2 Load contacts from file
 - 3.4 Other

- 3.4.1 Zoom in/out and scroll whenever necessary
- 4 Book Shelf Software
 - 4.1 Book management
 - 4.1.1 Load books from Download folder, provide support to .txt and .pdf
 - 4.1.2 Delete books
 - 4.1.3 Add category of books
 - 4.1.4 Manage categories (add/remove books)
 - 4.1.5 Search books by text query
 - 4.2 Book reading
 - 4.2.1 Swipe to go to the next/previous page
 - 4.2.2 Bookmark a page and go to the bookmark page (optional for .pdf)
 - 4.2.3 Day & night mode (optional for .pdf)
 - 4.2.4 Search for word and go to the word (optional for .pdf)
 - 4.2.5 Change font and size of the text in the book (optional)
 - 4.2.6 Extract chapters and directly go to certain chapters (optional)
 - 4.3 Book notations (optional)
 - 4.3.1 Add notation to certain page
 - 4.3.2 View notations on page with notations
 - 4.3.3 Edit notations
 - 4.3.4 View all notations for a book
 - 4.3.5 Delete notations
 - 4.4 Other
 - 4.4.1 Zoom in/out and scroll whenever necessary
 - 5 A shipping software
 - 6 A match making software (matching people to books/hobbies they like)
 - 7 A comprehensive smart phone application
 - 8 A scheduling software (an airline flight scheduler, course scheduler, task scheduler, etc.)
 - 9 Ticketing software such as in transportation (airline, train, cruise, etc.) domain or in social domain (culture center, athletic complex, hotel, etc.)
 - 10 E-commerce software
 - 11 Online banking software
 - 12 Non-for profit organization automation (e.g. library) software
 - 13 University Information System software (student registration, faculty, course schedules, ...)
 - 14 A rental facility software involving one or more of the following: vehicles, video, audio, books, games, and others)

- 15 A mobile application for suggesting a good match (for a restaurant, bookstore, healthcare provider, etc.) based on user preferences 16 ...

This is not a comprehensive list. We encourage students to brain storm and come up with original ideas other than what is listed below. We want everyone to enjoy this semester long project and contribute the most. Therefore, think broadly and choose a topic you will enjoy working on.

Some useful links are as follows:

UML Editors:

Following is a list of some freely available UML editors for your convenience:

- Sparks Enterprise Architect <http://www.sparxsystems.com/>. 30 day trial version only.
- Violet UML editor
<http://alexdp.free.fr/violetumleditor/page.php>. Very simple features of UML design. Free.
- Omondo EclipseUML <http://www.omondo.com/> (Academic License available for free) works with Eclipse <http://www.eclipse.org/>
- StarUML <http://staruml.sourceforge.net/en/> see also [StarUML @ Wikipedia](#)

Open-source UML modeling tool supports most of the diagram types specified in UML 2.0

- UMlet <http://www.umlet.com/>

Open-source UML tool; runs stand-alone or as an [Eclipse](#) plug-in on Windows, OS X, and Linux

- Visual Paradigm for UML (Community Edition)

<http://www.visualparadigm.com/product/vpuml/editions/community.jsp>

The Community Edition is free for non-commercial use; It puts a “Community Edition” watermark on your diagrams; Runs on Windows XP/Vista/7, Linux, Mac OS X, etc.

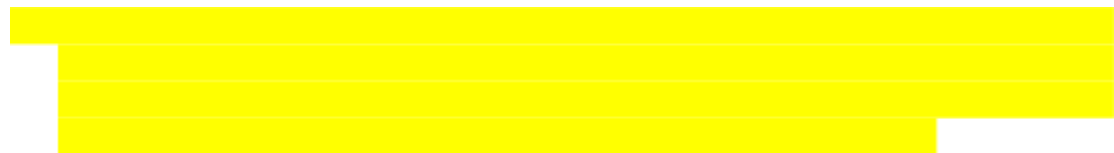
- Netbeans UML Plug-in <http://www.netbeans.org/features/uml/>

Does not support all UML diagram types, but supports forward and reverse engineering

- ArgoUML <http://argouml.tigris.org/> see also [ArgoUML @ Wikipedia](#)
- Rational Rose <http://www.rational.com/tryit/index.jsp>
- <http://www.microgold.com/>
- [Microsoft Visio](#) and open-source [Dia](#) are diagramming tools with a library of UML shapes that may also be used for drawing UML diagrams.
- Creatly <http://creately.com/> for drawing UML diagrams.

Making life easy when working as a group:

It is very important to make sure that you communicate and share common work with your teammates. Here are some URLs to help you on that:



- [Github](#) — a web-based Git or version control repository and Internet hosting service. This is the recommended version control software for this project. If by some reason you cannot use the Github platform, you may use any of the following similar platforms for sharing your project related material.
- [Doodle](#)—a tool for time management and meeting scheduling.
- [GroupMe](#)—a group messaging service that lets you be in touch with your team members via mobile phones.
- [CVS, open source version control](#) - helps you work on different versions of the same product and merge your versions.
- [Slack](#) — a web-based team communication service.
- Mercurial <https://www.mercurial-scm.org/> for version control.