Lab-3 Design and Diagrams

Expand Your User Stories

- Prioritize your user stories created from last lab.
- Expand 2/3 (team of three /four) / 4 (team of five) user stories.
- Do it together!

Why Expanding Your User Story?

- Improve clarity, more detailed descriptions of the user's needs and requirements).
- Better alignment, ensure everyone has a shared understanding of the goals, priorities…
- Provide a more comprehensive understanding of the requirements and ensure everyone is working towards the same objectives.

How to Expand Your User Story?

- **Select a user story you want to expand.**

  1. with the greatest impact and deliver the most value first
  2. complex, require more detail for the team to understand
  3. important to stakeholders
  4. technically feasible and can be delivered within a reasonable timeframe.

How to Expand Your User Story? (cont.)

- **Definition of “done”. (optional)**

  1. define the specific conditions that must be met for the user story to be considered complete
  2. make sure the conditions are clear, concise and easily understandable
  3. don’t skip any edge cases or exceptional scenarios

How to Expand Your User Story? (cont.)

- **Provide context.**
  1. include relevant information about the user and his/her environment (personas)
  2. name & role

How to Expand Your User Story? (cont.)

- **Describe the behavior and flow of the system.**

1. identify interactions between the user and your web app (e.g., click a button, fill out forms)
2. map out the flow of interactions (start with the user's first interaction and continue through to the end)
3. define the steps, including what the user does, what your system does in response, any condition must be met before proceed.
4. document all steps and interactions in a clear manner.

How to Expand Your User Story? (cont.)

- **Break down into subtasks. (optional)**

  1. make sure you have a clear and complete understanding of the selected user story
  2. identify the subtasks / decide which specific steps need to be completed to deliver the user story
  3. assign tasks to team members based on their skills, experience and availability / decide who is responsible for each task

An Expanded User Story

- **Persona(s)**

  David, media consumer

- **User Story**

  As a media consumer, I want to be able to add basic information about the media I consume (e.g., title, media type, author, etc.), so that I can keep a record of what I've consumed

- **Definition of done (optional)**
  - *a user can add records to the database and view a list of all records*
  - *all tasks have been completed (developed, tested, reviewed, and validated)*

An Expanded User Story (cont.)

- **Steps / interactions**
  - *start application*
    - *show home screen*
  - *click "add record"*
    - *show form to add information*
  - *type in information and click "submit"*
    - *propagate data to data service / DB*
    - *retrieve all records*
    - *redirect to records listing*

- **Tasks (optional)**
  - *UX designs / user testing*
  - *backend / data management*
  - *frontend / UI development*
  - *feature / integration testing*

Check Lab Doc for More Info

- Lab docs are uploaded to Canvas (Assignments)
- References/examples

Tools / resources:

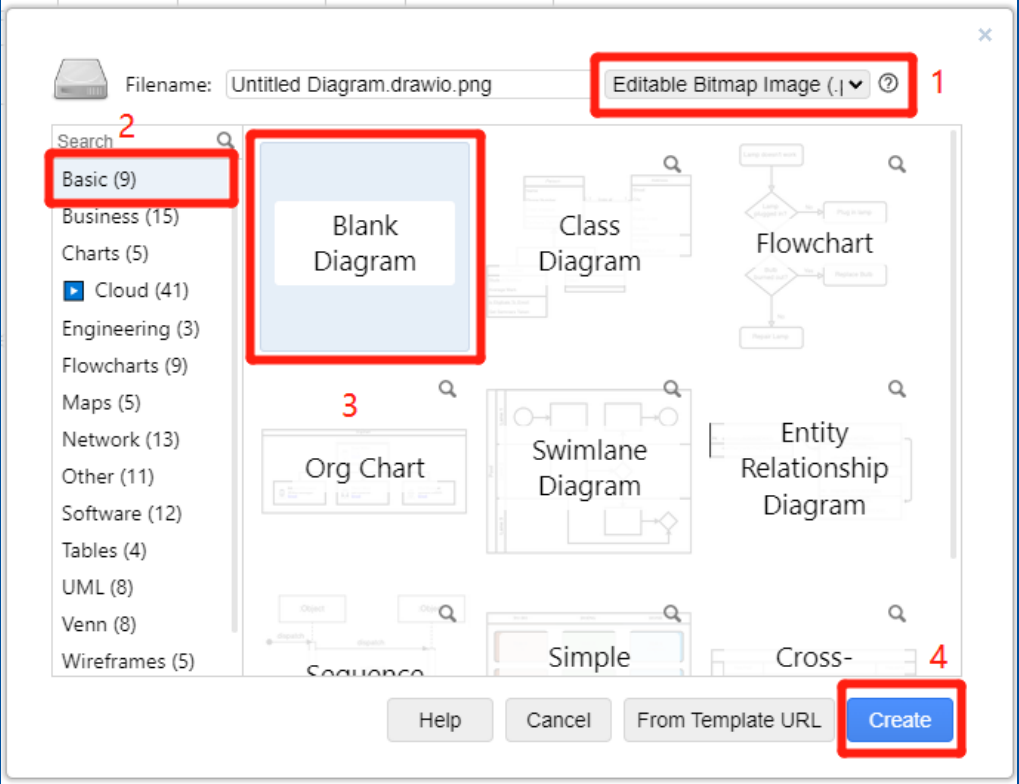
- notes:
  - cs518.lab2a.releases-stories NOTES [extended user stories]
  - cs518.lab2b NOTES
- For design diagramming, I tend to use diagrams.net

Diagrams

- Architectural diagram * 1 (team)
- Sequence diagram * size of your team (individual)
- Create diagrams w/ diagrams.net (formerly known as draw.io)

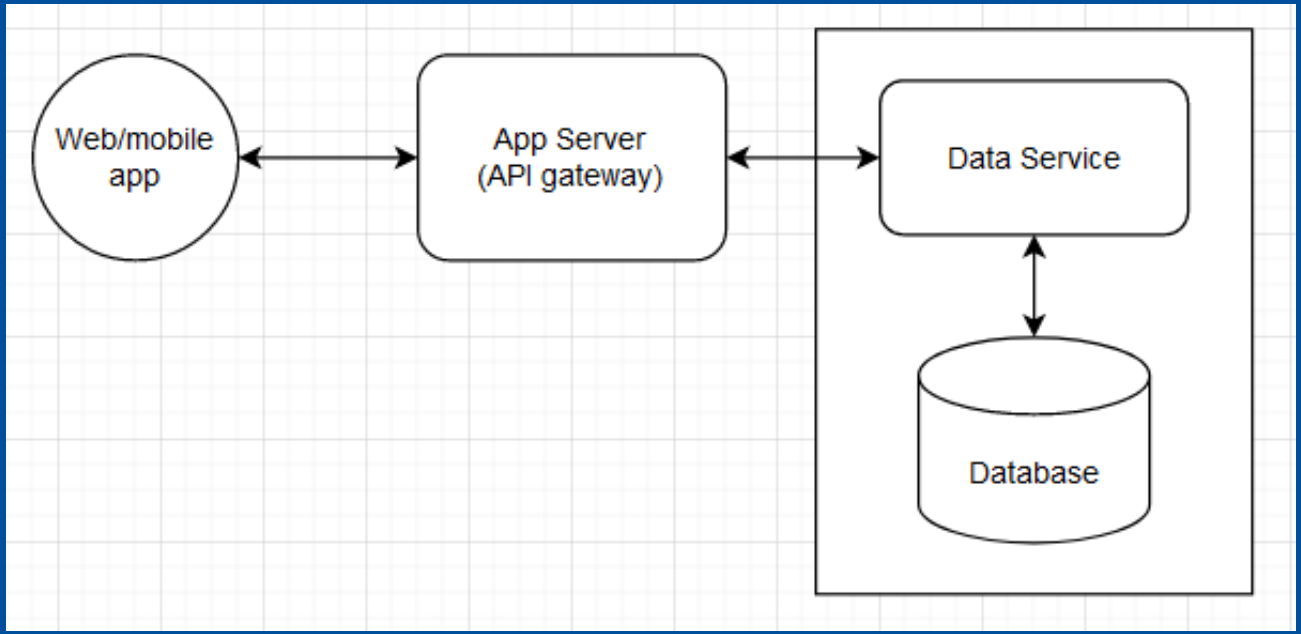
Create a New Architectural Diagram

- diagrams.net is an open-source technology stack for building diagramming applications, it is purely browser-based.



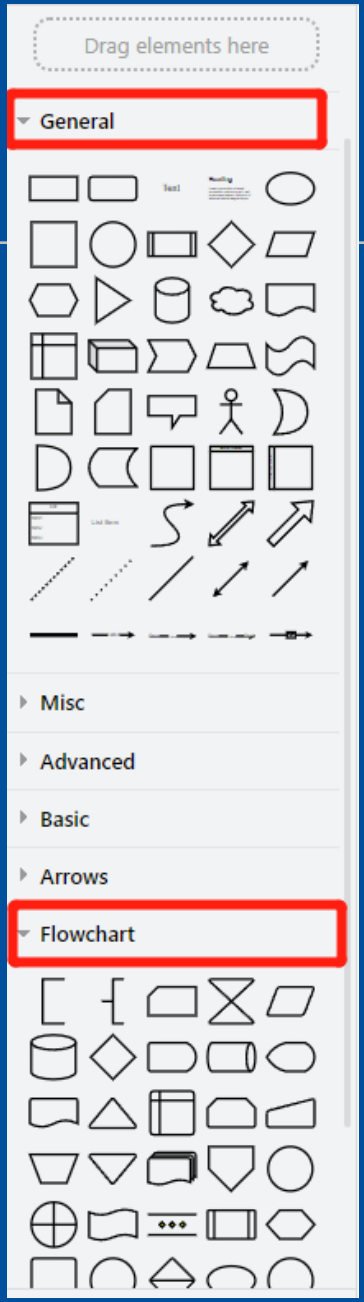
Architectural Diagram

- Build your architectural diagram based on the following generic microservice architecture.



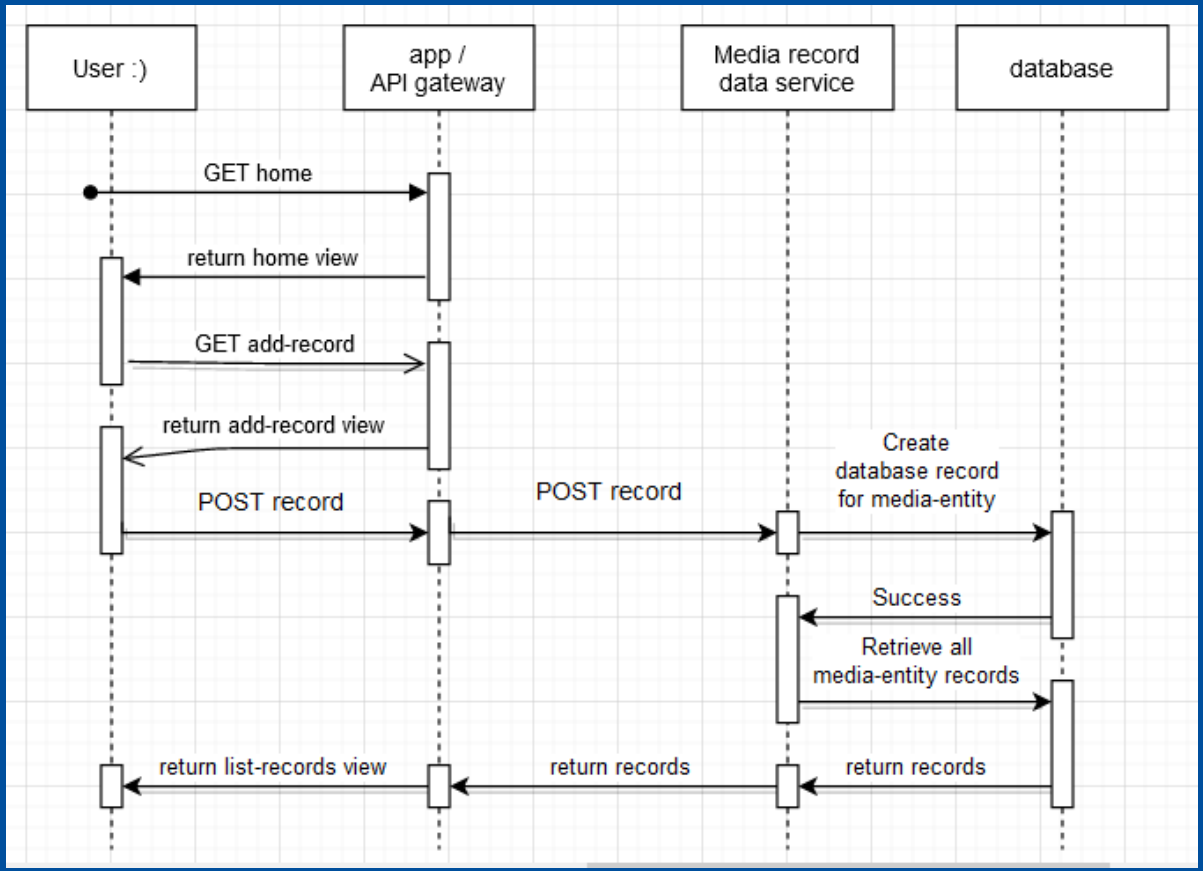
Shapes

- Circle
- Rounded rectangle
- Database
- Text
- Bidirectional connector

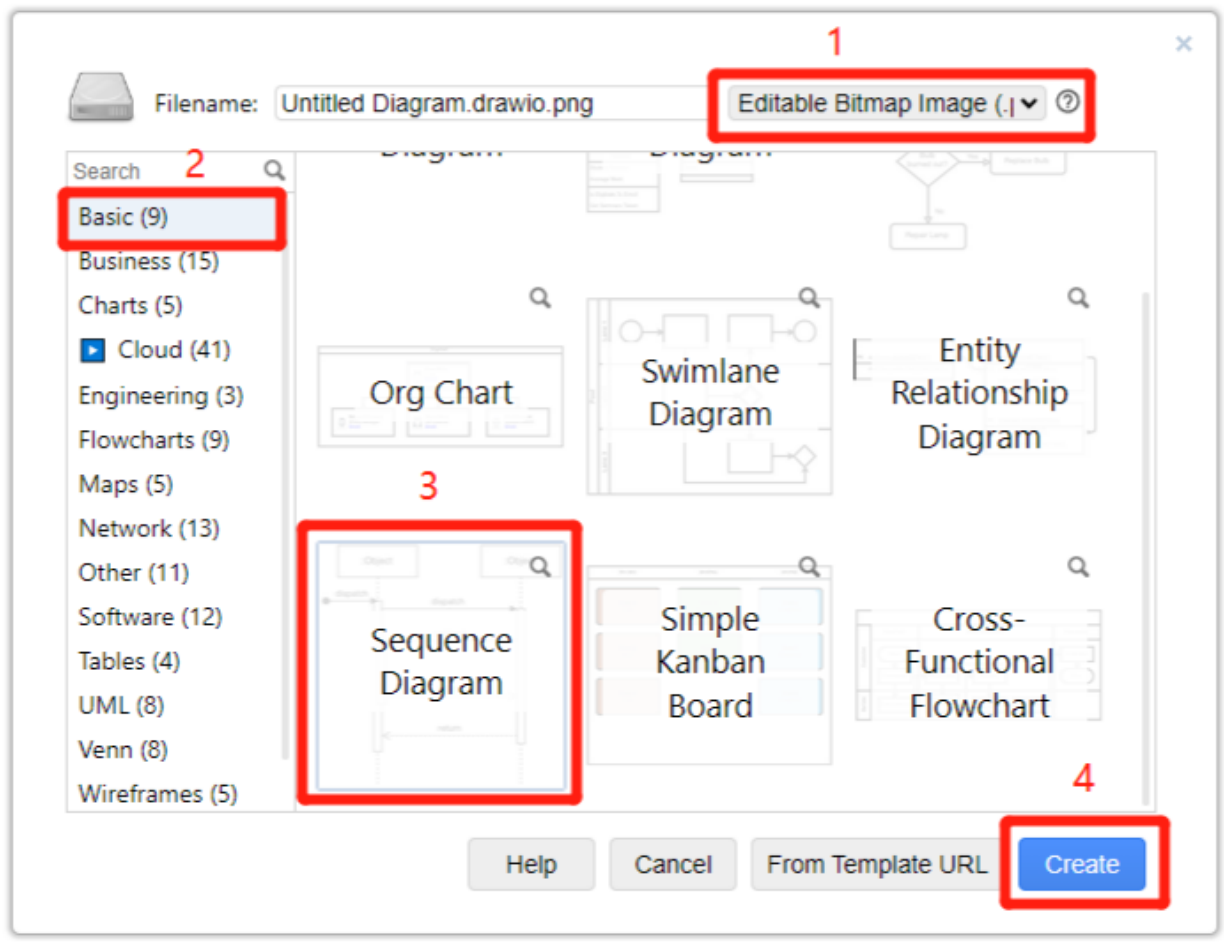


Sequence Diagram

- Illustrate how a group of objects interact and operate with each other sequentially

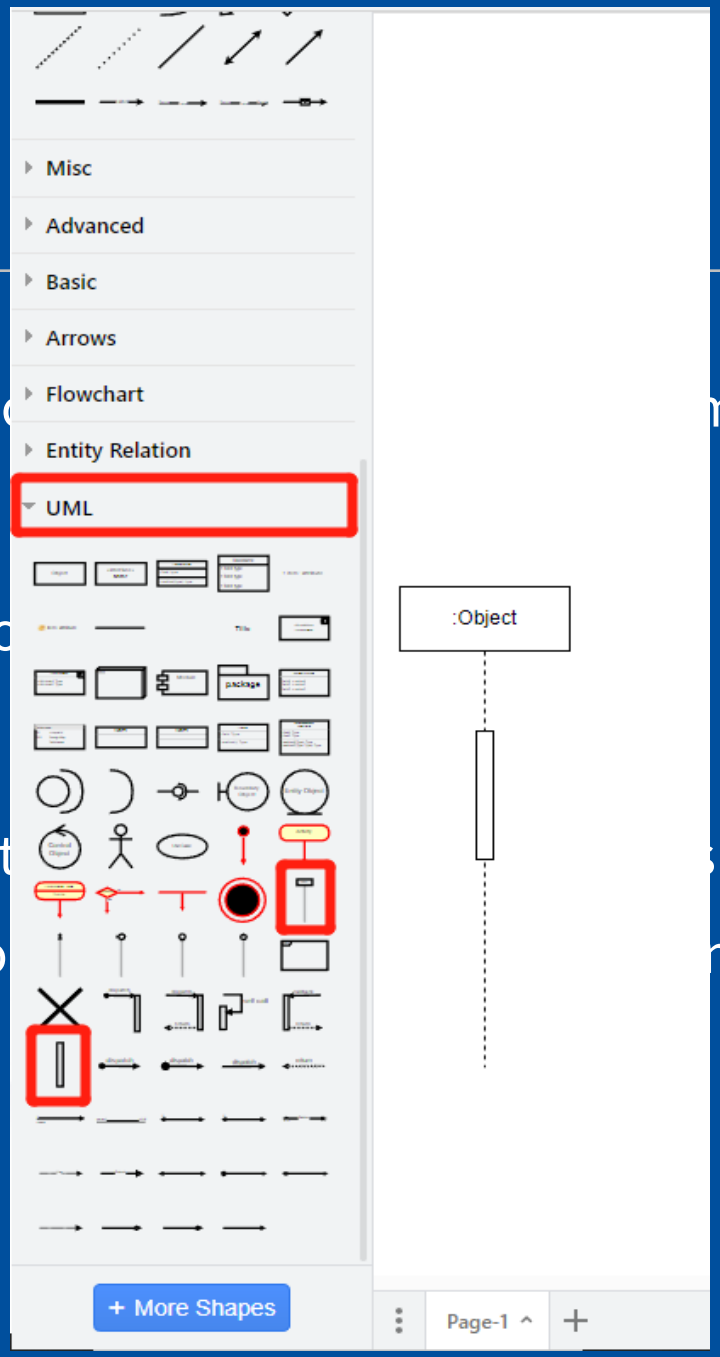


Create a New Sequence Diagram



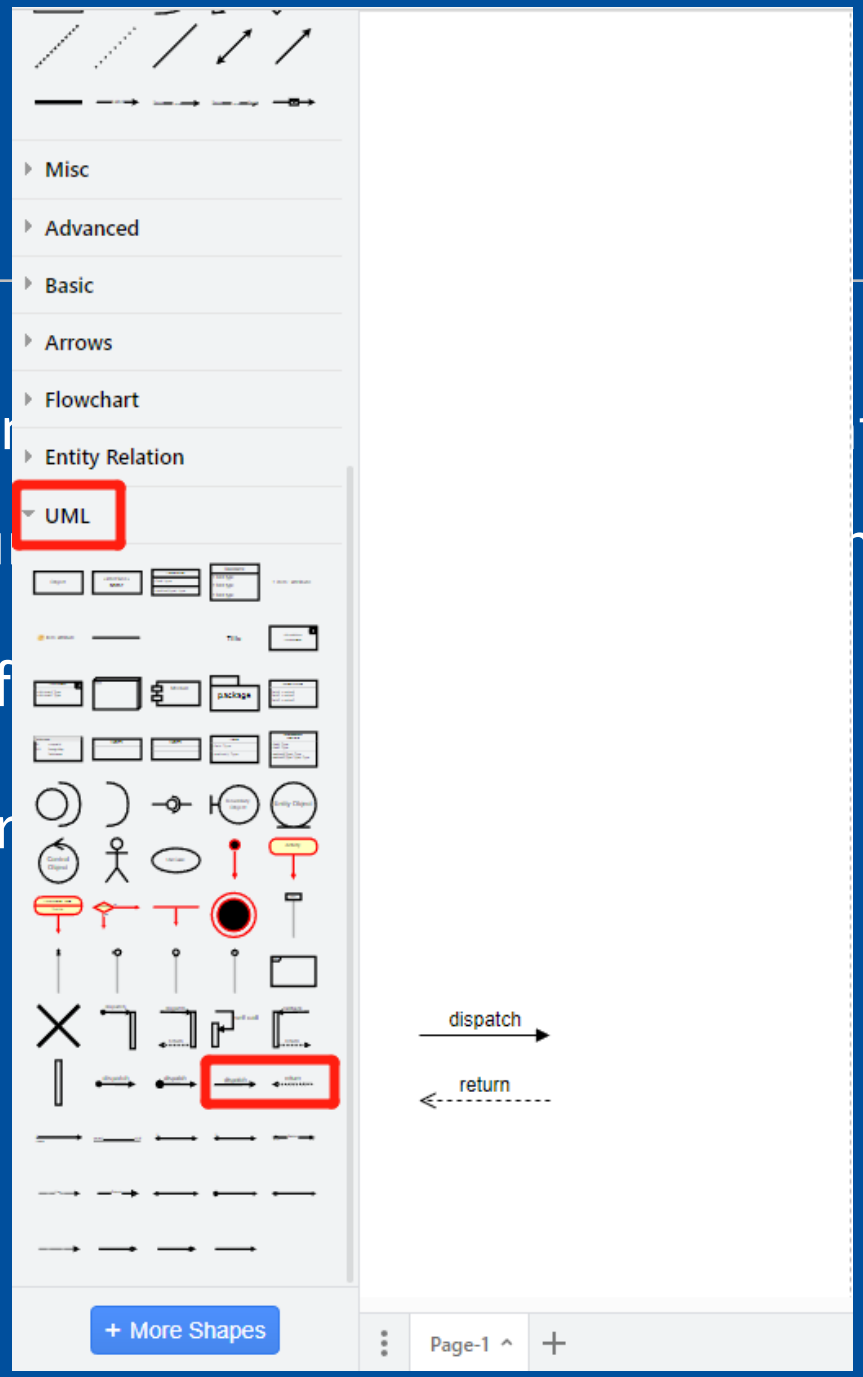
UML Shapes

- Object, represents a specific instance of a me written inside).
- Lifeline, represents interactions between o (vertical dashed line).
- Activation bar, represents the duration of t the operation or action (rectangle, height is pro me that the object performs the action)



UML Shapes (cont.)

- Message, represents a communicatio ts in the system, typically in the form of a fu n.
- Return, represents the return value of
- Messages and returns are shown as ar



Tips and Tricks

- Start with a clear plan.
- Make sure you have the right object selected before you start dragging/moving/deleting.
- Connect to the right object.
- Make use of the undo and redo buttons.
- Make use of the *Search Shapes* function.

Submission (due 2/21 11:59 p.m.)

- A document (docx / pdf) to Canvas, INDIVIDUAL submission

  one person -> arch diagram

  the rest -> sequence diagrams + corresponding expanded user stories

- GitLab

  add your document to your team's GitLab repository

More Questions?

?