Development Specs

CS 485/698: AI-Assisted SE

Today’s Agenda

- Team meeting (~20 minutes)
  - Retrospective on P1, plan for P2
- Use an LLM to turn Wednesday’s user stories into development specifications
  - We’ll try mobbing again

Today’s Agenda

- Team meeting (~20 minutes)
  - Retrospective on P1, plan for P2
- Use an LLM to turn Wednesday’s user stories into development specifications
  - We’ll try mobbing again

Course Announcement:
Michael’s OH are moved this week to Tuesday, 11:30-12:30 (same place as usual)

Team Meeting

- Spend the next ~20 minutes with your team
  - Discuss what went well or poorly during P1. Are there any changes to your work process that you want to make before P2? (~10 minutes)
  - Who is responsible for what for P2? (~10 minutes)
  - If you finish these, start P2
- Michael ~~and I~~ will be coming around to discuss your P1 submissions

Development Specifications

Development Specifications

- Motivation: we need to turn our user stories into features we can code into our app

Development Specifications

- Motivation: we need to turn our user stories into features we can code into our app
- Those features need to be connected by a ***software architecture***
  - Architecture and design are the "***glue***" between ***what your software is supposed to do*** and ***the code you actually write***

Development Specifications

- Motivation: we need to turn our user stories into features we can code into our app
- Those features need to be connected by a ***software architecture***
  - Architecture and design are the "***glue***" between ***what your software is supposed to do*** and ***the code you actually write***
- Every feature is described by a development specification

Development Specifications

- Motivation: we need to turn our user stories into features we can code into our app
- Those features need to be connected by a ***software architecture***
  - Architecture and design are the "***glue***" between ***what your software is supposed to do*** and ***the code you actually write***
- Every feature is described by a development specification
- A developer can be guided by a dev spec to build exactly what is necessary to make the feature work

Development Specifications

- Motivation: we need to turn our user stories into features we can code into our app
- Those features need to be connected by a ***software architecture***
  - Architecture and design are the "***glue***" between ***what your software is supposed to do*** and ***the code you actually write***
- Every feature is described by a development specification
- A developer can be guided by a dev spec to build exactly what is necessary to make the feature work
- Our goal today: figure out how to get an LLM to help us write these

How to turn a user story into a dev spec?

How to turn a user story into a dev spec?

- A user story answers ***why***

How to turn a user story into a dev spec?

- A user story answers ***why***
- In contrast, a dev spec answers ***how***

How to turn a user story into a dev spec?

- A user story answers ***why***
- In contrast, a dev spec answers ***how***

- Pair together and think about what details you would include for a dev spec for this user story (2 minutes!)

  As a grandma calling her grandson, I can't accidentally mute or turn off the call so that I don't think the call "broke."

How to turn a user story into a dev spec?

- A user story answers ***why***
- In contrast, a dev spec answers ***how***

- Pair together and think about what details you would include for a dev spec for this user story (2 minutes!)

  As a grandma calling her grandson, I can't accidentally mute or turn off the call so that I don't think the call "broke."

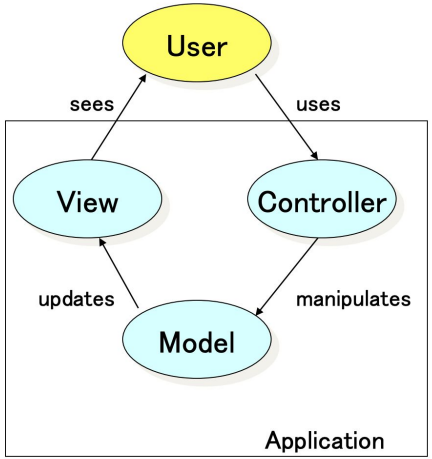
- Could a dev build it for you now? Anything missing?

Components of a Development Spec

- let’s look at the handout together

Model-view-controller (“MVC”)

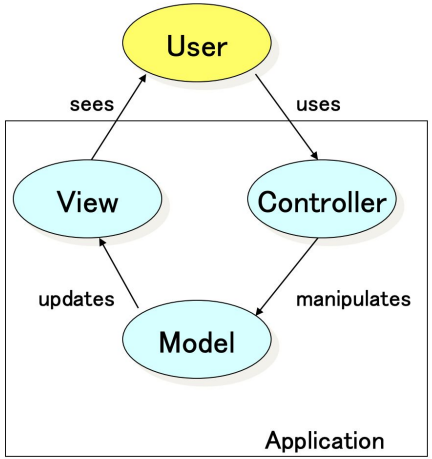
Definition: a ***model-view-controller architecture*** splits the project into three parts:



Model-view-controller (“MVC”)

Definition: a ***model-view-controller architecture*** splits the project into three parts:

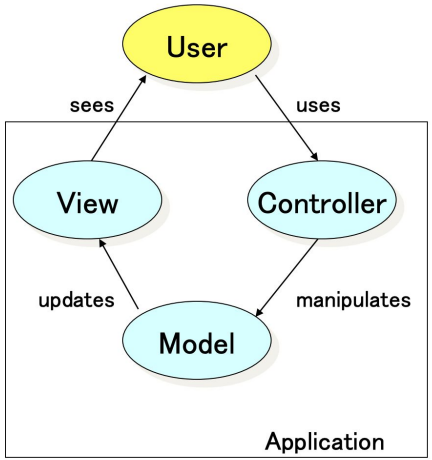
- a single ***model***, which is the application's dynamic data structure, independent of the user interface



Model-view-controller (“MVC”)

Definition: a ***model-view-controller architecture*** splits the project into three parts:

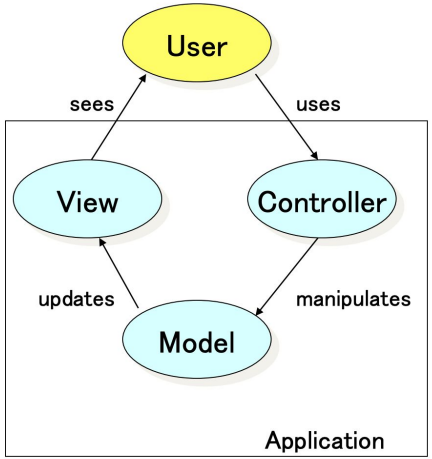
- a single ***model***, which is the application's dynamic data structure, independent of the user interface
- one or more ***views***, which are representations of information (e.g., charts, tables, or UIs)



Model-view-controller (“MVC”)

Definition: a ***model-view-controller architecture*** splits the project into three parts:

- a single ***model***, which is the application's dynamic data structure, independent of the user interface
- one or more ***views***, which are representations of information (e.g., charts, tables, or UIs)
- one or more ***controllers***, which accept input and convert it to commands for the model or view

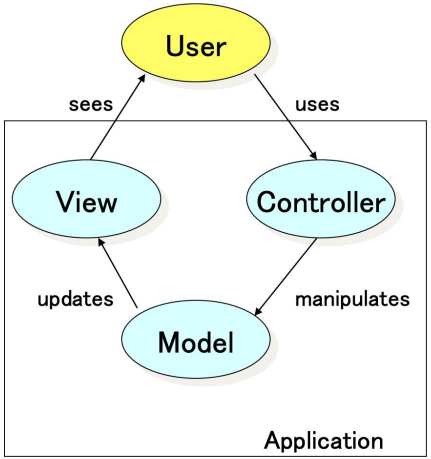


Model-view-controller (“MVC”)

Definition: a ***model-view-controller architecture*** splits the project into three parts:

- a single ***model***, which is the application's dynamic data structure, independent of the user interface
- one or mo
of inform
- one or m
convert i

Key advantage of MVC:

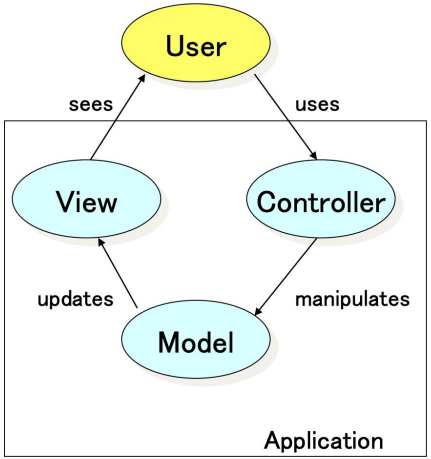


Model-view-controller (“MVC”)

Definition: a ***model-view-controller architecture*** splits the project into three parts:

- a single ***model***, which is the application's dynamic data structure, independent of the user interface
- one or mo
of inform
- one or m
convert i

Key advantage of MVC: **separates** data representation (Model), visualization/user interface (View), and client interaction (Controller)



Mobbing Activity

- link to mobbing sheet
- Let's get an LLM to make part of the dev spec for Zoom w/ better muting
- Everyone pick one good user story from Wednesday's class and copy it into the #mobbing channel.
- Which user story shall we specify today?
    - Mobber chooses, with input from the class
- Now, everyone tell the mobber what to say to their LLM to get it to create a dev spec for just this user story, section by section

Mobbing Activity, Part 2

- Mobber and notetaker swap
- Let’s focus now on threat modeling.
  - What are the technical risks we should worry about in our spec?
  - What are the privacy risks?
  - What are the security risks?
  - What are the risks that could prevent us from completing this feature?