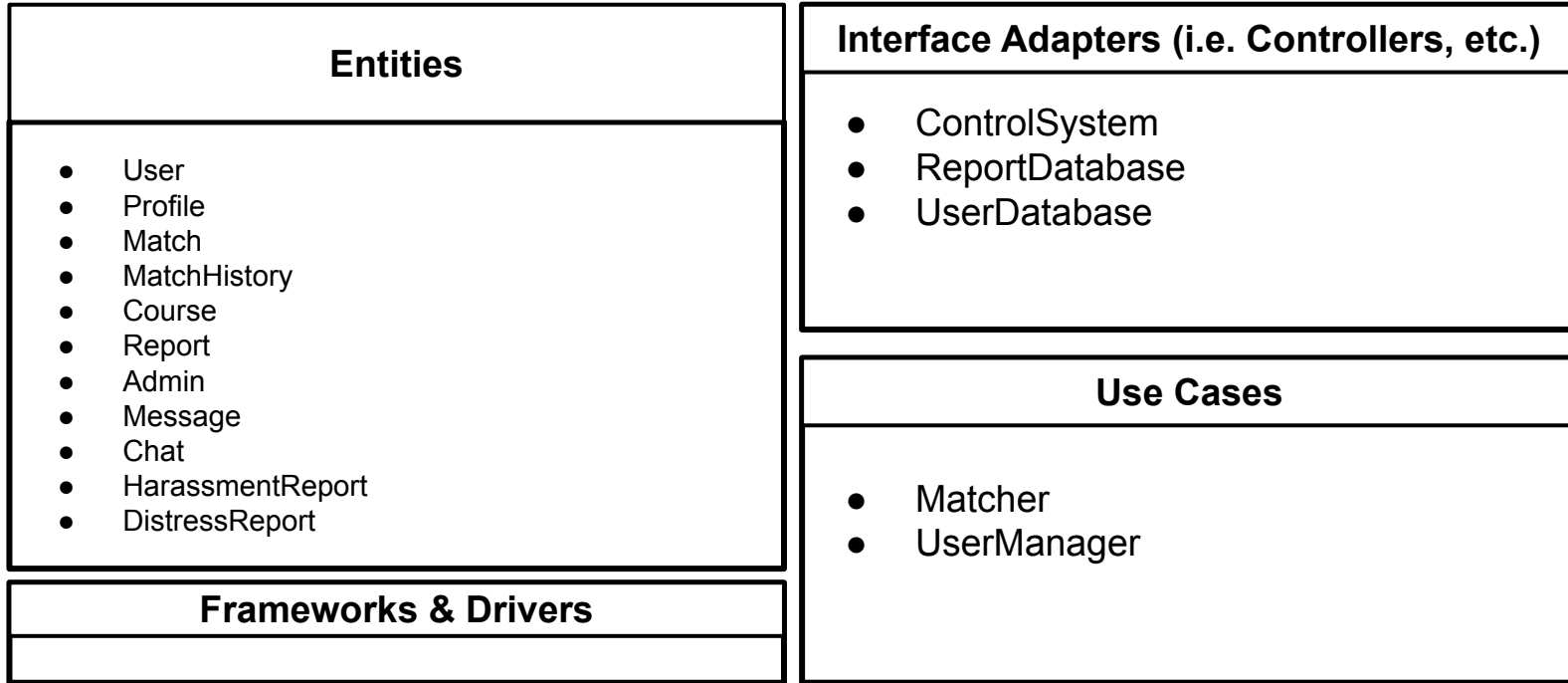


# Amigos CRC Model - Clean Architecture Overview



# CRC Card: ControlSystem

## ControlSystem (Controller)

- Retrieves dictionary of user ids mapped to user objects from UserManager and sends the dictionary to Matcher
- Receives and processes input
- Sends user id and/or instructions to UserManager and Matcher
- Receives matches from Matcher and sends it to UserManager

Matcher  
UserManager

# CRC Card: UserManager

## UserManager (Use Case)

- Creates and deletes user and user profile
- Modifies user profile using profile setters
- Retrieves a dictionary which maps user ids to user objects from UserDatabase
- Stores the dictionary when the program is running
- Sends the dictionary to ControlSystem
- Receives matches from ControlSystem
- Modifies User's current matches

User  
UserDatabase  
Profile  
ControlSystem

# CRC Card: User

User (Entity)		Admin
<ul style="list-style-type: none"><li>• Stores a profile (using an instance method)</li><li>• Stores a unique identifier (users are unique)</li><li>• Stores current matches</li><li>• Stores MatchHistory</li><li>• Flag (a boolean) for report</li><li>• Stores BlockedUser (users that this person blocks)</li><li>• Stores checkbox (a boolean) for wildcard (random) matches</li></ul>	Profile MatchHistory	

# CRC Card: Profile

## Profile (Entity)

- Stores information (private attribute) regarding
  - Name
  - Year of Study
  - Program of Study
  - Courses for a Term
  - Interest(s)
  - Contact Information
- Have getters for all of these
- Setters for all of these

Course

# CRC Card: Course

Course (Entity)	
<ul style="list-style-type: none"><li>• Stores the course code</li><li>• Stores their preferred lecture/tutorial/section</li><li>• Stores all the lecture sections they can attend</li></ul>	User

# CRC Card: UserDatabase

## UserDatabase (Gateway)

- Writes and reads users to and from external storage (database/text file)
- Sends list of users to UserManager

UserManager

# CRC Card: Matcher

Matcher (Use Case)	
<ul style="list-style-type: none"><li>• Generates matches based on user's courses and interests</li><li>• Generates wildcard (random) matches for a user if that user if that user has opted for it</li><li>• Sends matches to ControlSystem for it to handle how to store matches</li></ul>	ControlSystem



# CRC Card: Match

Match (Entity)	
<ul style="list-style-type: none"><li>• Stores the two users' IDs</li><li>• Date of match</li><li>• Stores the Metric/Percentage of Match</li></ul>	User

# CRC Card: MatchHistory

## MatchHistory (Entity)

- Has a 'clean' method to erase rejected history for the past term
- Stores the amigos per user
- Stores rejected matches per user
- Stores blocked users

User  
Match

# CRC Card: Report

<b>Report (Entity)</b>	
<b>HarassmentReport</b> <b>DistressReport</b>	
<ul style="list-style-type: none"><li>• An abstract class which contains report information for an admin to review and resolve</li><li>• Stores the following information<ul style="list-style-type: none"><li>○ Which user filed the report</li><li>○ Which user is the report filed against</li><li>○ Date of report filing</li><li>○ Has the admin resolved the report (boolean)</li></ul></li></ul>	

# CRC Card: ReportDatabase

ReportDatabase (Gateway)	
<ul style="list-style-type: none"><li>• Stores reports sorted by time filled</li></ul>	Admin

# CRC Card: HarassmentReport

HarassmentReport (Entity)		Report
<ul style="list-style-type: none"><li>• It is a subclass of report meant only for harassment cases</li></ul>		

# CRC Card: DistressReport

DistressReport (Entity)		Report
<ul style="list-style-type: none"><li>• Stores optional report information<ul style="list-style-type: none"><li>○ Stores user id of the user in distress<ul style="list-style-type: none"><li>■ Displays distress hotline to the user in distress</li></ul></li></ul></li></ul>		

# CRC Card: Message

Message (Entity)	
<ul style="list-style-type: none"><li>• Stores the content of the message</li><li>• Stores the sender of the message</li><li>• Stores the date and time of message</li></ul>	User

# CRC Card: Chat

Chat (Entity)	
<ul style="list-style-type: none"><li>• Stores an ordered collection of messages</li><li>• In a chat, each message can only have one out of two senders</li></ul>	User



# CRC Card: Admin

<b>Admin (Entity)</b>		<b>User</b>
<ul style="list-style-type: none"><li>• Admin is a User but with special privileges</li><li>• Admin can access reports through the ReportDatabase and mark a report as resolved</li></ul>	ReportDatabase	