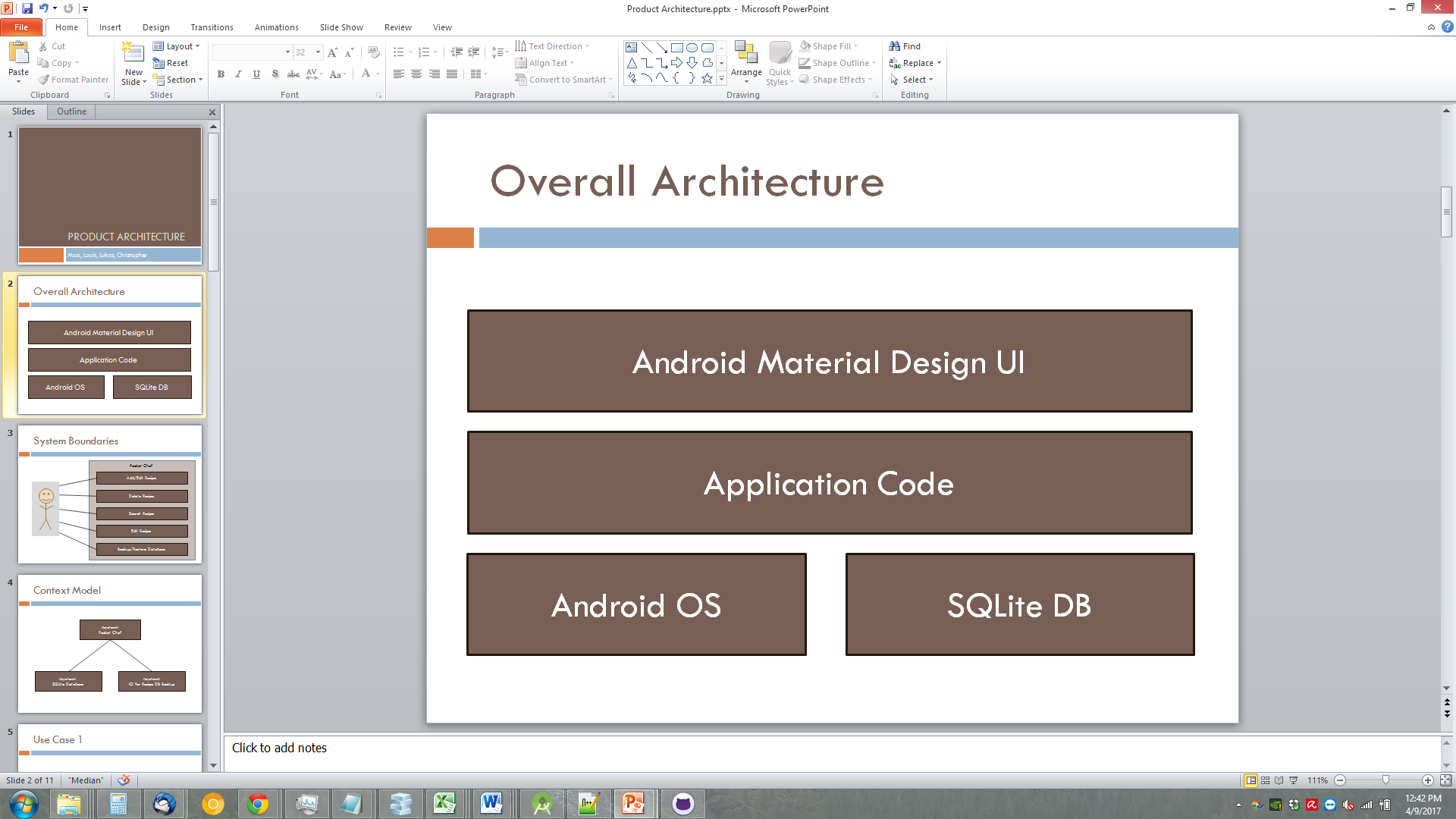
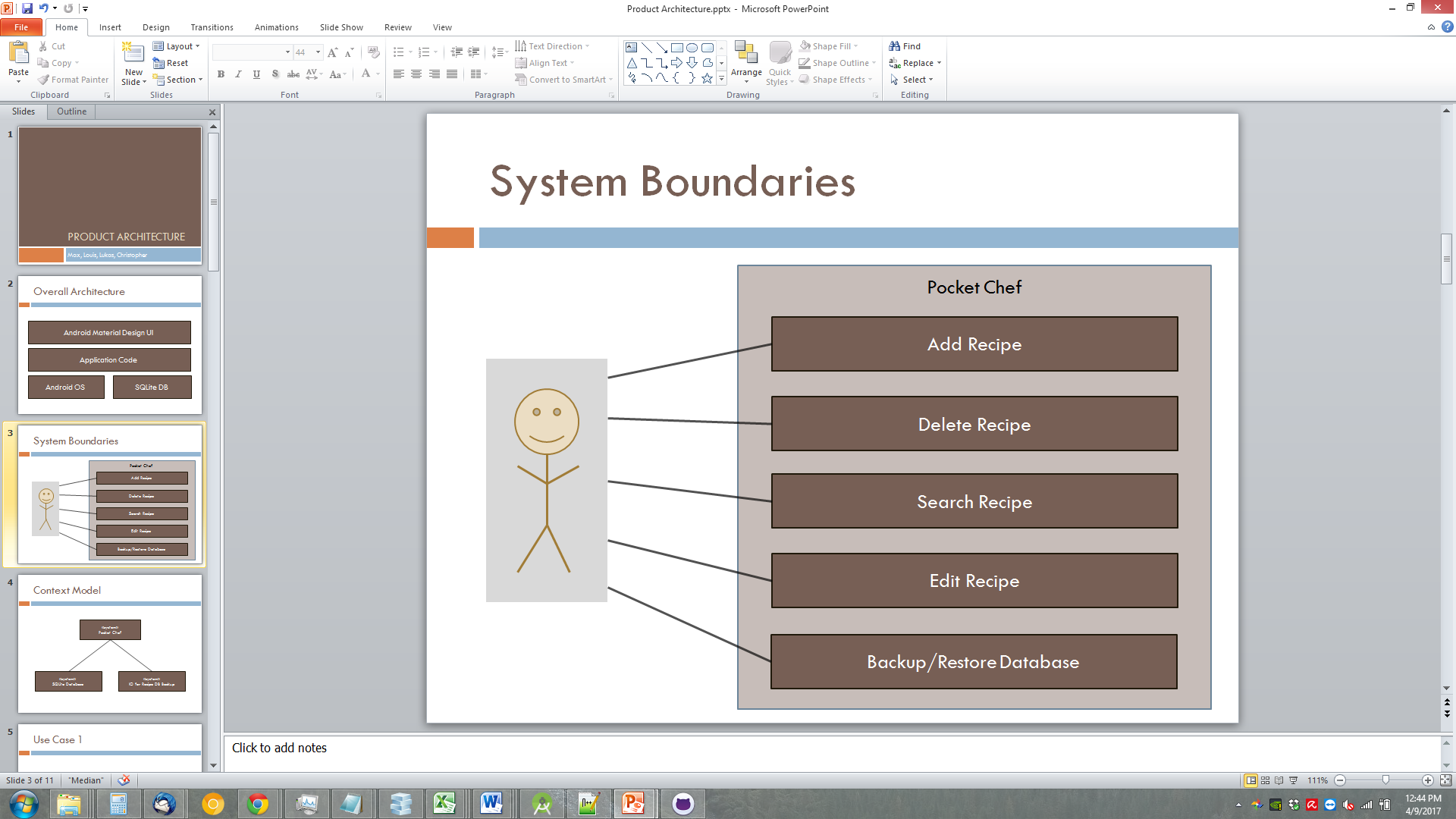
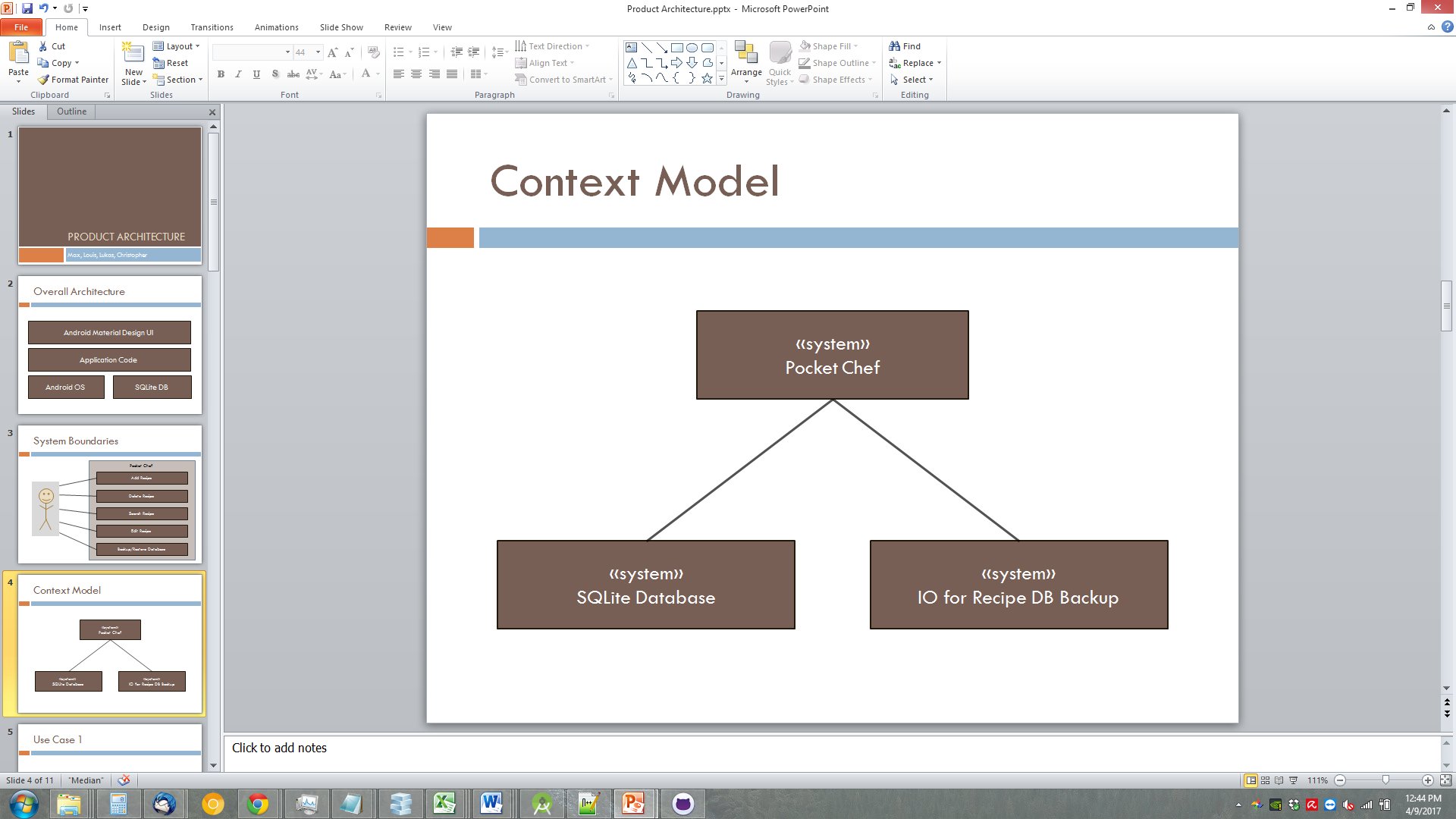
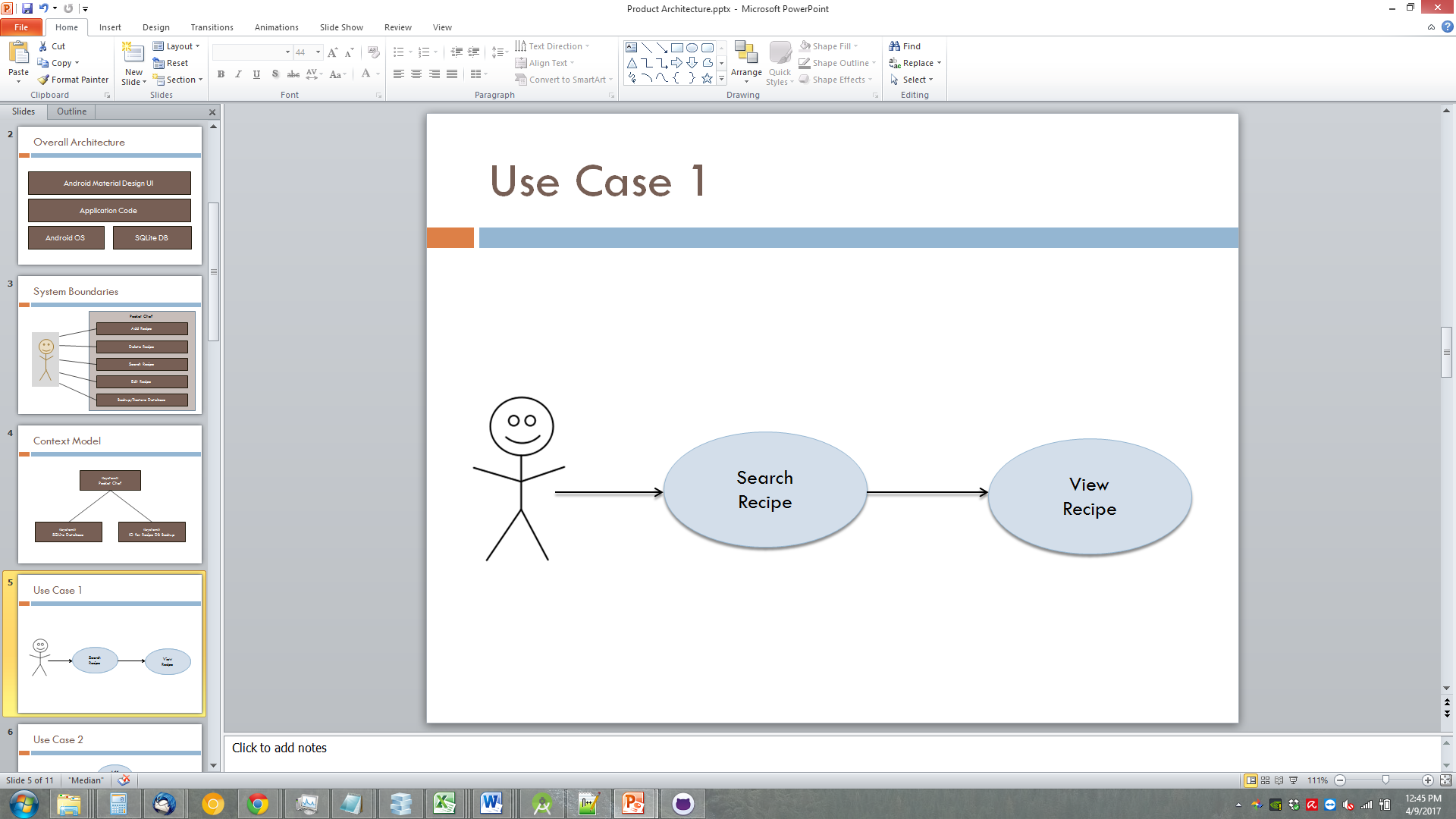
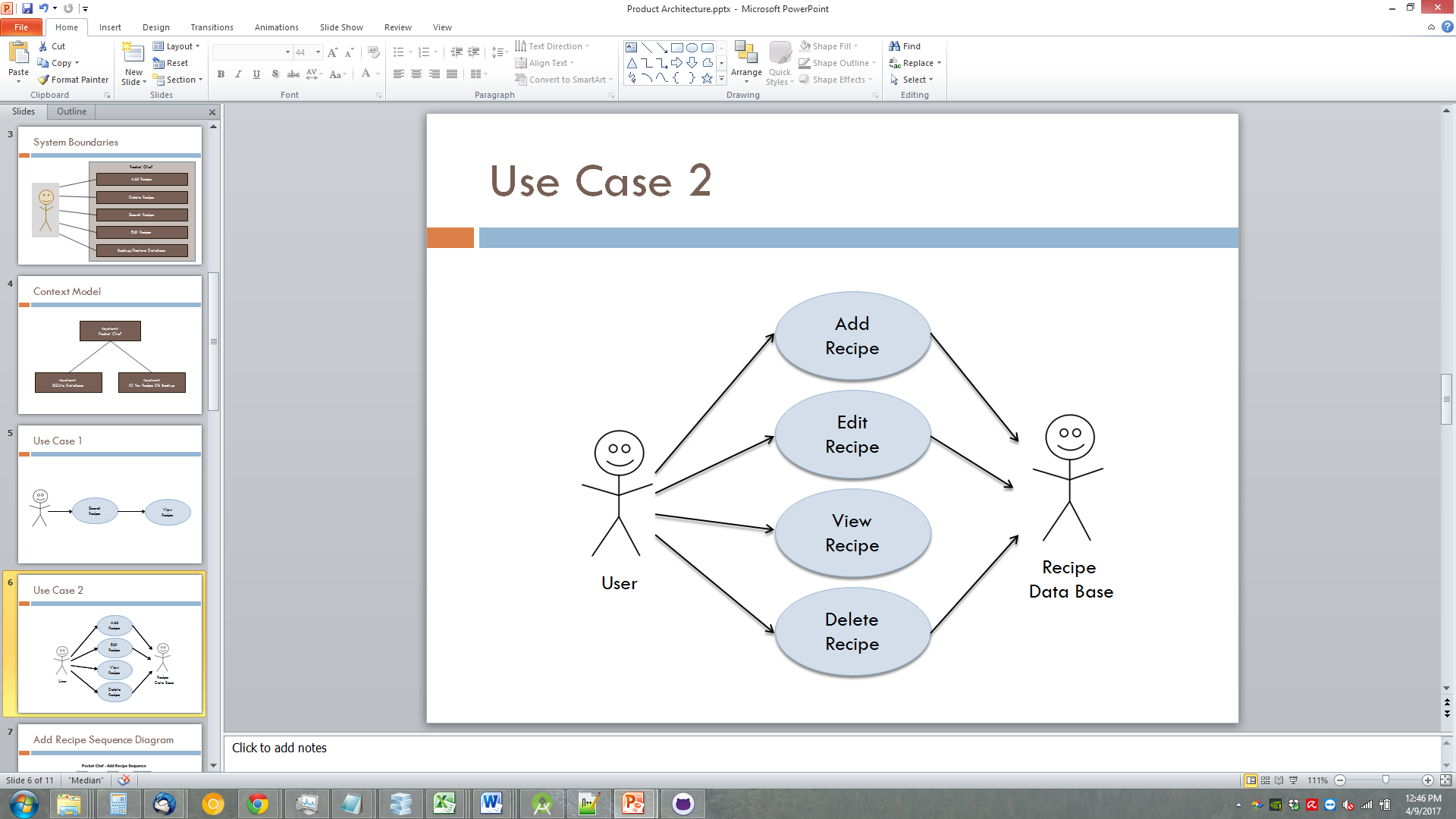
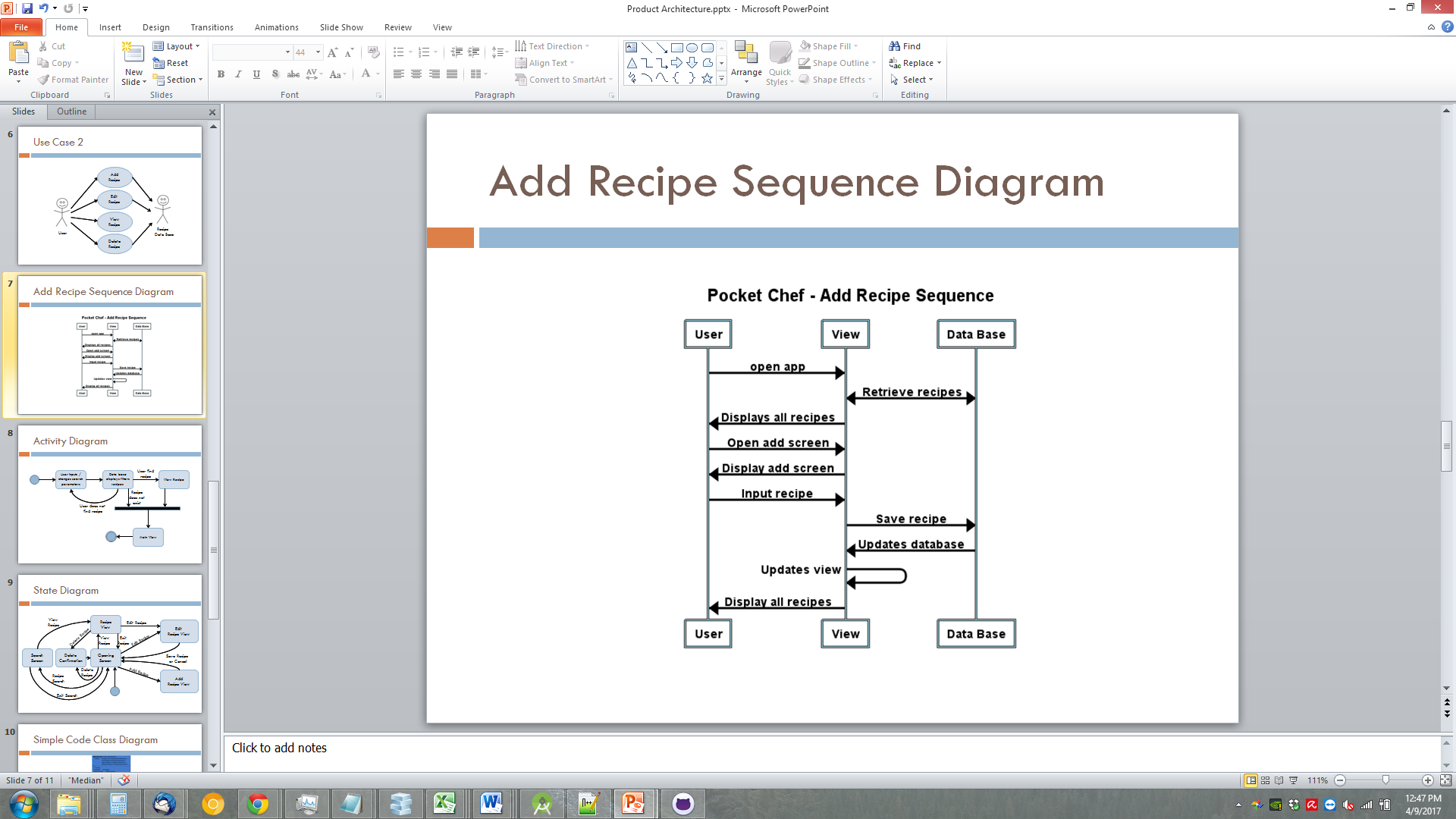
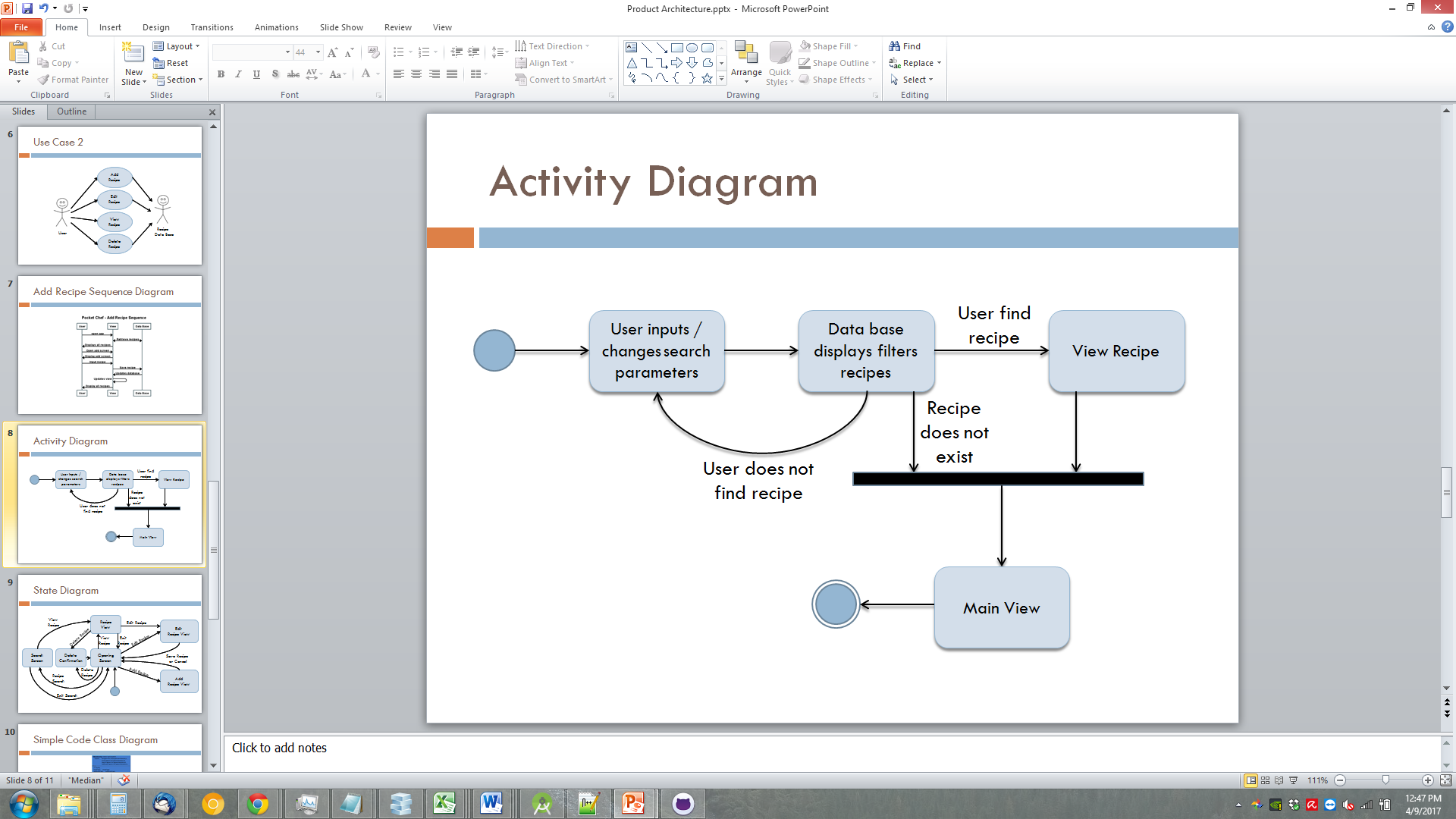
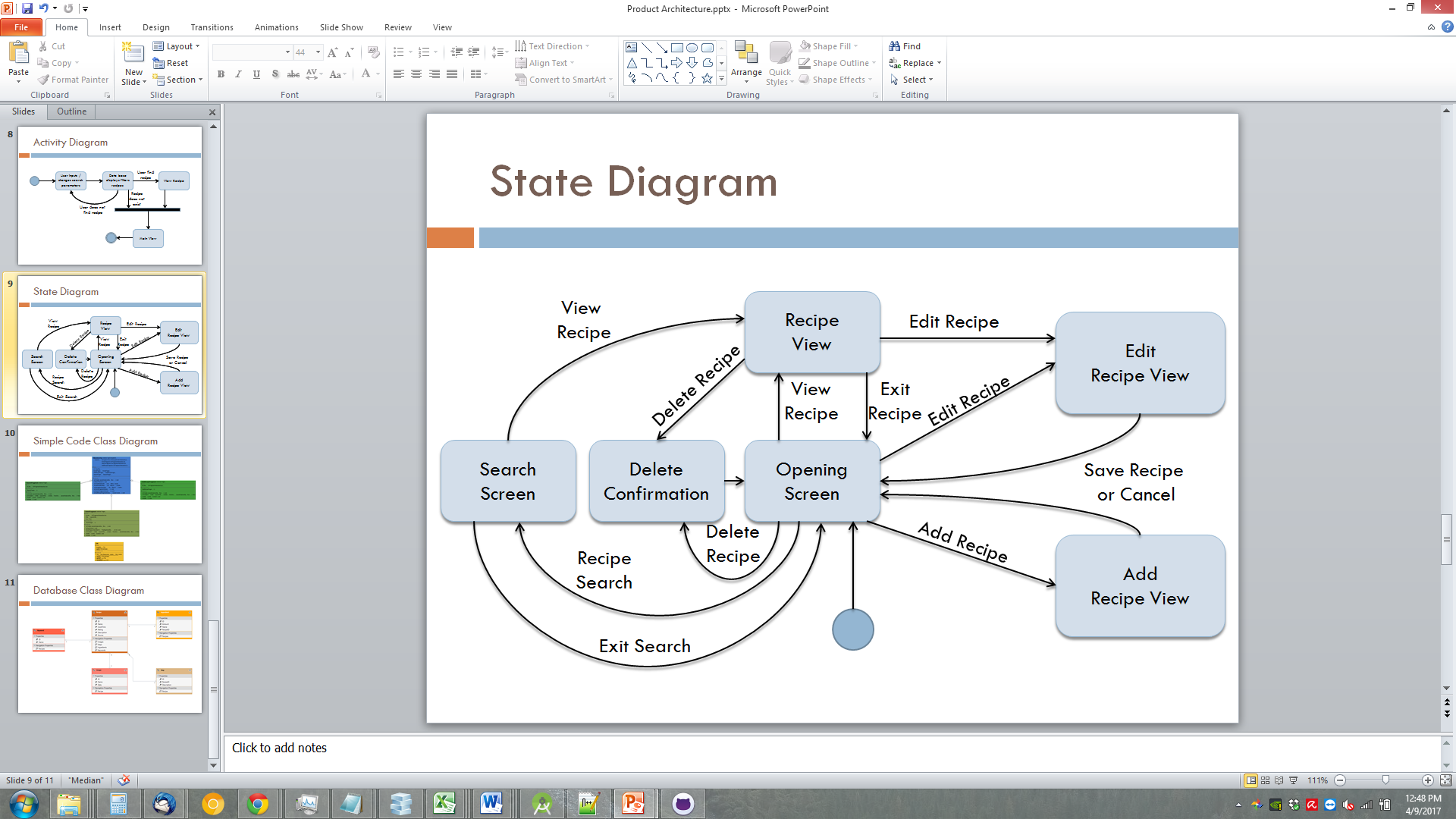
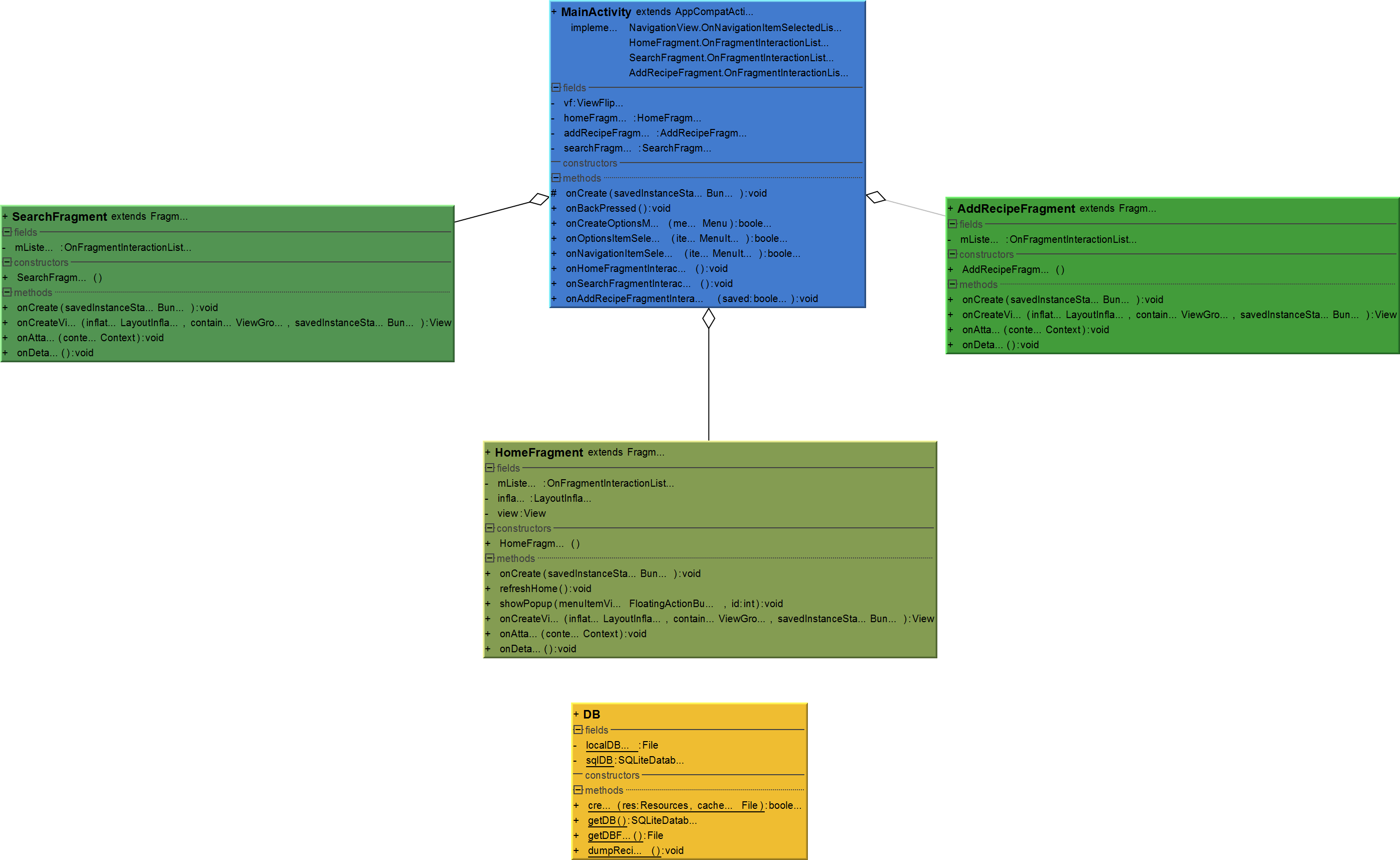
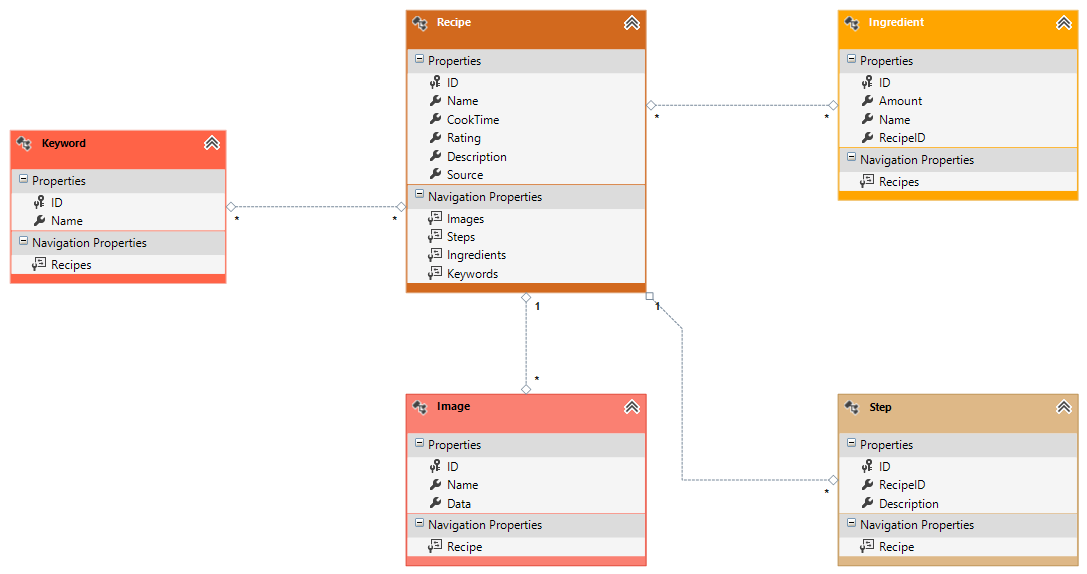
Product Architecture

* Overall Architecture
  + 
  + The visual above shows how we have split up our system architecture.
  + Our Architecture follows a layered format. Our application has dependences on lower-level components.
* System Boundaries
  + 
  + Our system boundaries as shown above demonstrate the scope and organization of our application’s structure.
  + Each component (box) represents a use case in our application.
* Context Model
  + 
  + Our context model above shows external systems that our application relies on to function.
* Use Case I
  + 
  + This use case shows how a user might find a specific recipe by searching through the use of keywords and phrases.
* Use Case II
  + 
  + This use case shows how a user might interact with the recipe database through the application.
  + Each component (in the middle) shows how many portions of the application depend on the use of the database.
* Add Recipe Sequence Diagram
  + 
  + The Add Recipe Sequence shown above walks through how our application handles adding a recipe to the database.
  + We first show all available recipes, then show an add recipe screen, and then once the new recipe is added to the database, we refresh the view, essentially completing the cycle.
* Activity Diagram
  + 
  + The activity diagram above shows how an application search activity is completed through the process of different phases.
* State Diagram
  + 
  + The above state diagram shows the states the application can be in, and how you can transition between states.
  + Our main stages of the state diagram can be represented as different fragments (UI screens) in our application.
* Simple Code Class Diagram
  + 
  + The above class diagram shows how our main classes interact with each other, as well as the different fields associated with each class.
  + Our classes shown in different shades of green can be seen as different portions of our UI, and the blue class can be seen as the activity that handles the UI for our application.
* Database Class Diagram
  + 
  + Our database class diagram, as shown above, shows the different tables located in our database, as well as the different fields (columns) in each of those tables.
  + We also show the kind of relationship that our tables have with each other, such as many-to-one and many-to-many.