System Design Document

To be submitted to the Department of Mathematics and Computer Science, Gordon College
in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science

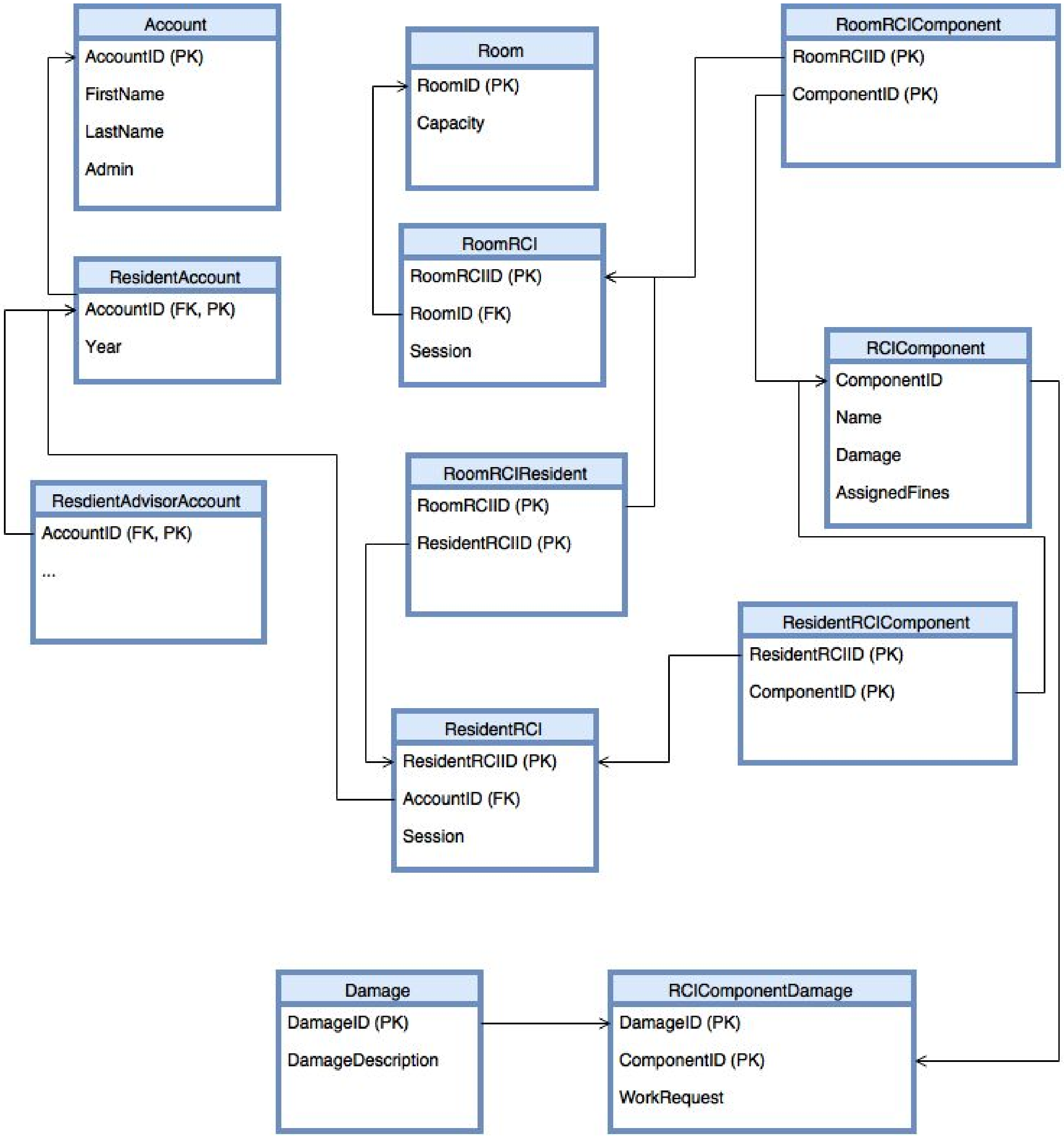
by:
Eze Anyanwu,
Stephanie Powers,
Weiqiu (Rachel) You

Document accepted on ___________ by _____________________________________
(date) (client)

Document accepted on ___________ by ______________________________________
(date) (departmental representative)

System Design Document

Database Schema



Notes:

The Views that have been provided for us before-hand are not represented here.

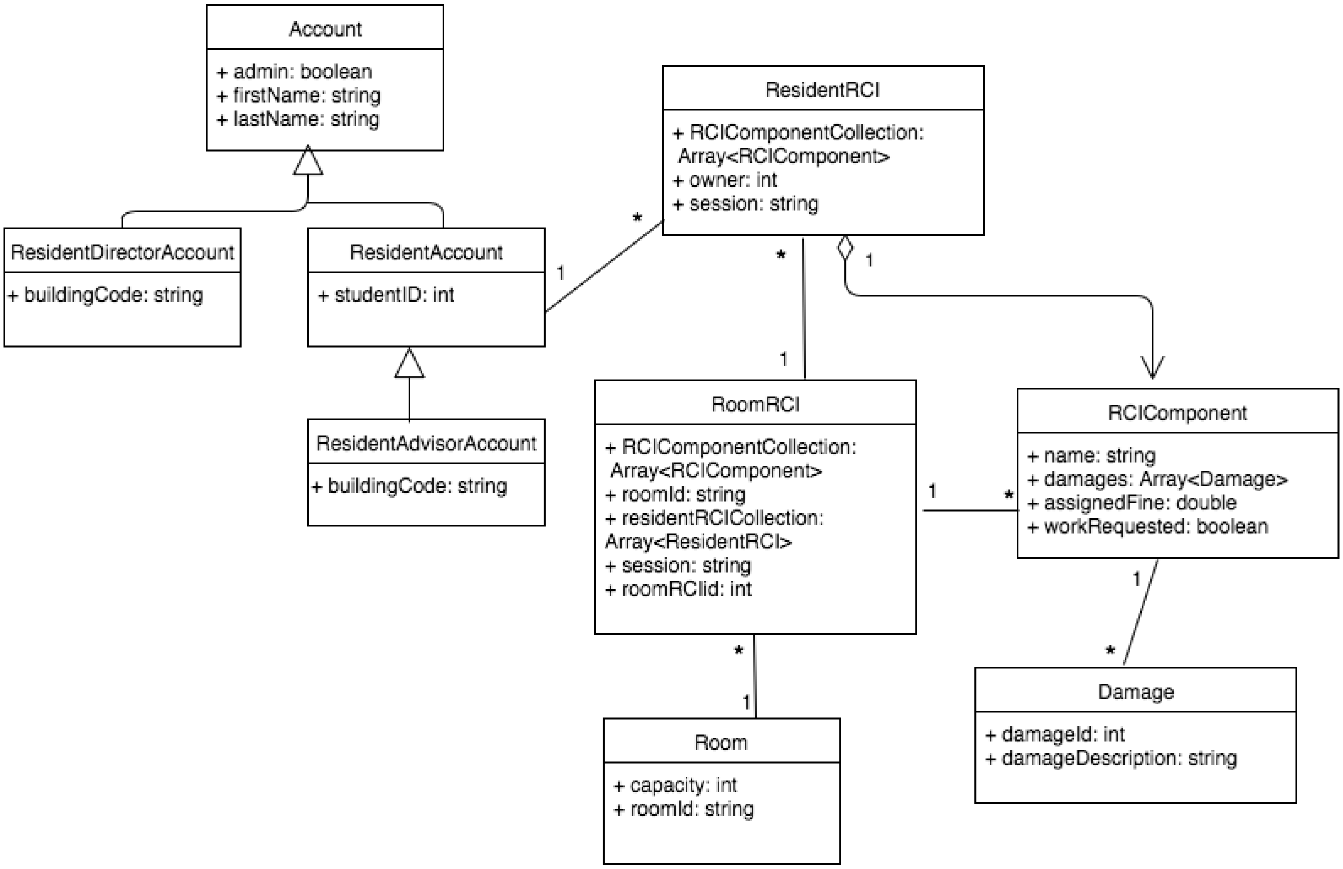
The `ResidentAdvisorAccount` table uses `ResidentAccount’s` primary key as its own primary key to model the concept that a Resident Advisor must be a Resident first and not just an ordinary Account.

The `RoomRCIComponent` and `ResidentRCIComponent` tables are used to model the collections of RCI components that belong to a `RoomRCI` and `ResidentRCI` respectively.

The `RoomRCIResident` table is used to model the collection of resident RCIs that belong to a `RoomRCI.`

An RCI component can have multiple type of damages (e.g. A wall can have paint chip, stain and a huge gash). To represent this, we use the `RCIComponentDamage` table model the collection of damages that belong to a specific `RCIComponent.`

Class Diagram



Class Descriptions:

- `Account`: represents a user as an account object, with firstName and lastName string attributes, as well as the boolean admin to indicate if an user should be given admin privileges or not

- `ResidentDirectorAccount` : inherits from `Account` and has a buildingCode attribute to denote what building the RD should have access to in the system

- `ResidentAccount` : inherits from `Account` and has a studentId attribute

- `ResidentAdvisorAccount:` Inherits from `Resident Account,` given that RA's are also residents themselves. Like the RD, RA's will have a buildingCode attribute denoting what building they should have access to

- `ResidentRCI:` represents a single resident's collection of room components during a single year in a single dorm. This object strongly relates to the current RCI paper form that residents fill out. Every `ResidentAccount` could have potentially many `ResidentRCI` objects associated with it, from year to year or even within a semester.

- `RCIComponent` : represents each element within a room (e.g. desk, carpet) and its corresponding collection of damages and fines, as well as the boolean workRequested to represent whether or not a workRequest has been filed for a given component.

- `RoomRCI`: allows room components to belong just to the room, particularly relevant in the case of an RA pre-filling RCI's for freshmen who have not yet moved in, to claim which room components belong to whom.

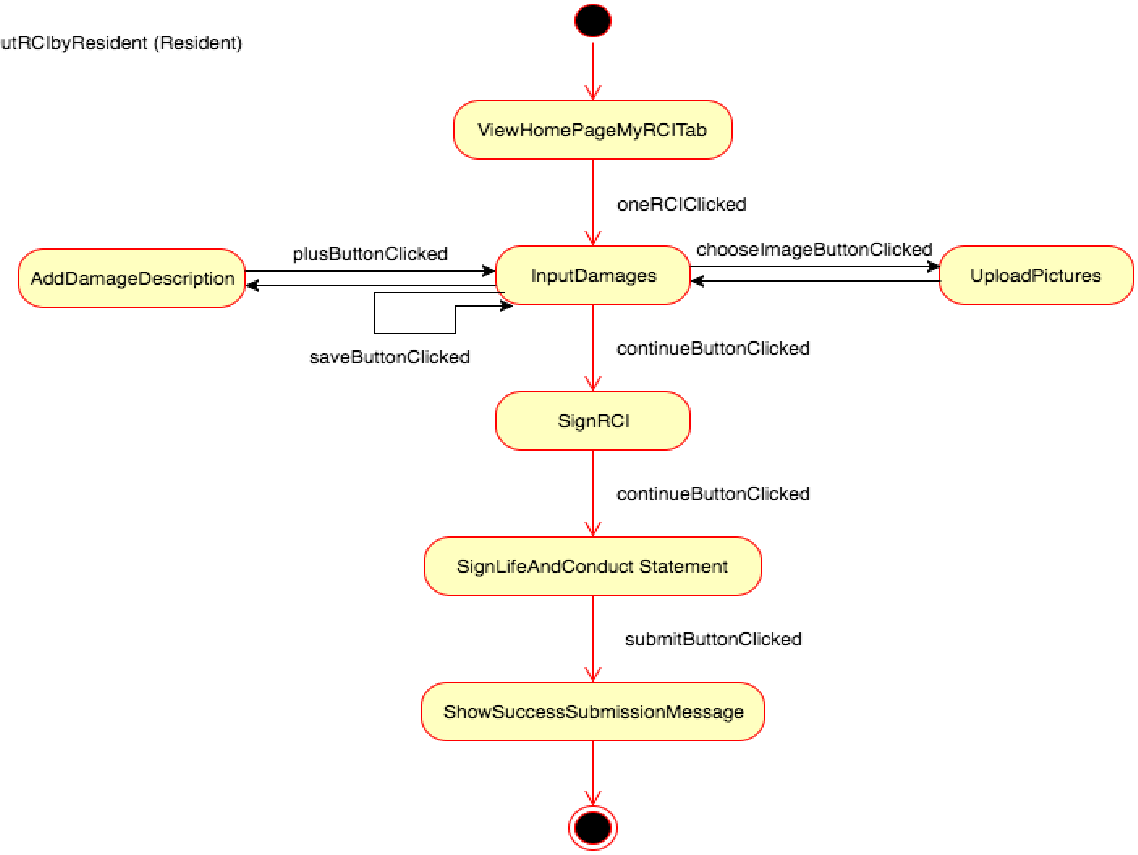
- `Room`: represents the physical room, which stays relatively constant. Enables us to keep a history of all the `RoomRCI` objects associated with a given room.

- `Damage`: represents an individual damage for an individual component. Particularly useful in the case in which a component may have multiple damages.

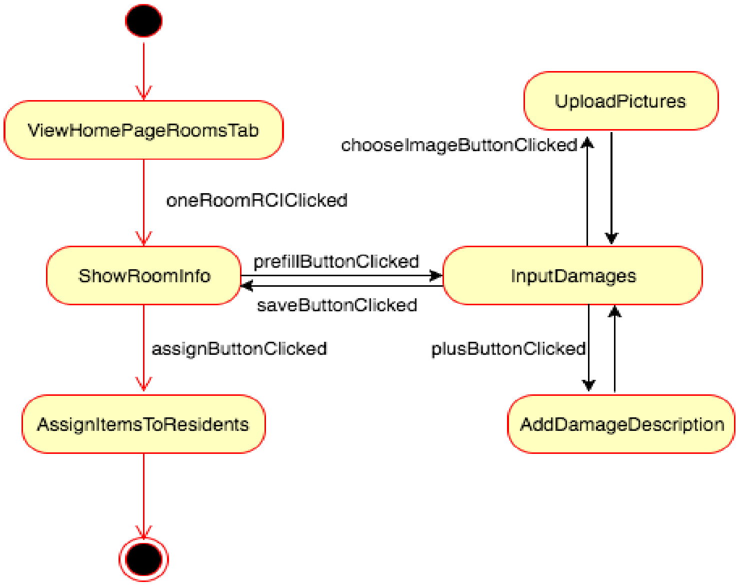
State Diagram

The six state diagrams below each represents one use case.

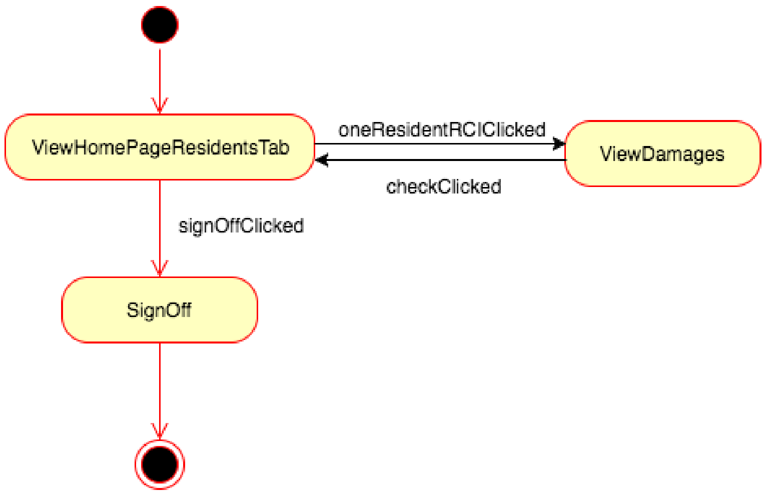
FillOutRCIbyResident (Resident)



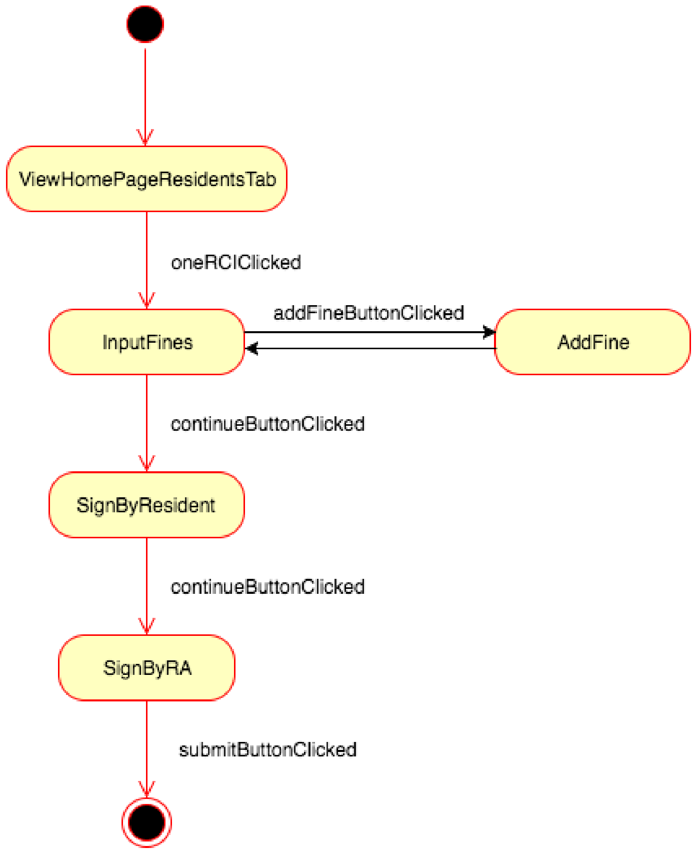
PreFillRCIforFirstYearStudent (RA)



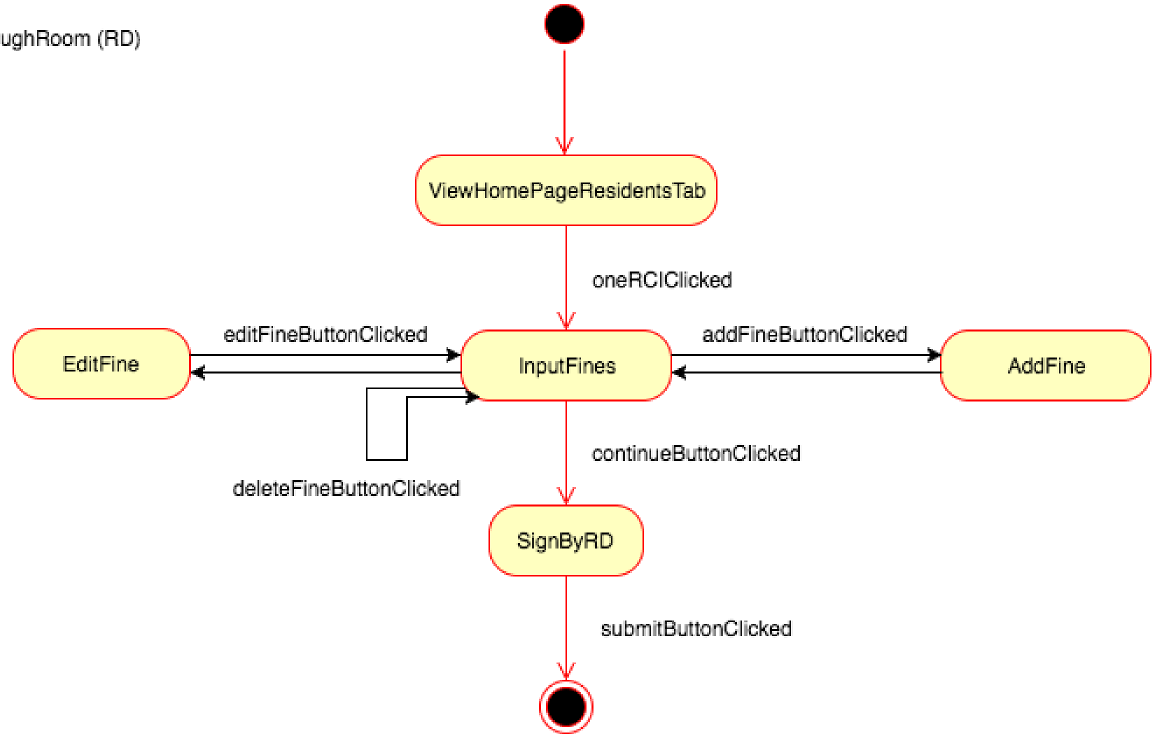
ReviewRCIbyRD (RD)



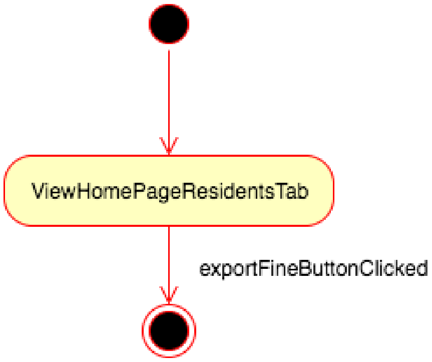
CheckOutResident (RA, with Resident)



WalkThroughRoom (RD)



GenerateFinesReport (RD)


ViewHomePageResidentsTab
exportFineButtonClicked