

Figure 7: Domain Model of the Plan a Family Activity Use Case

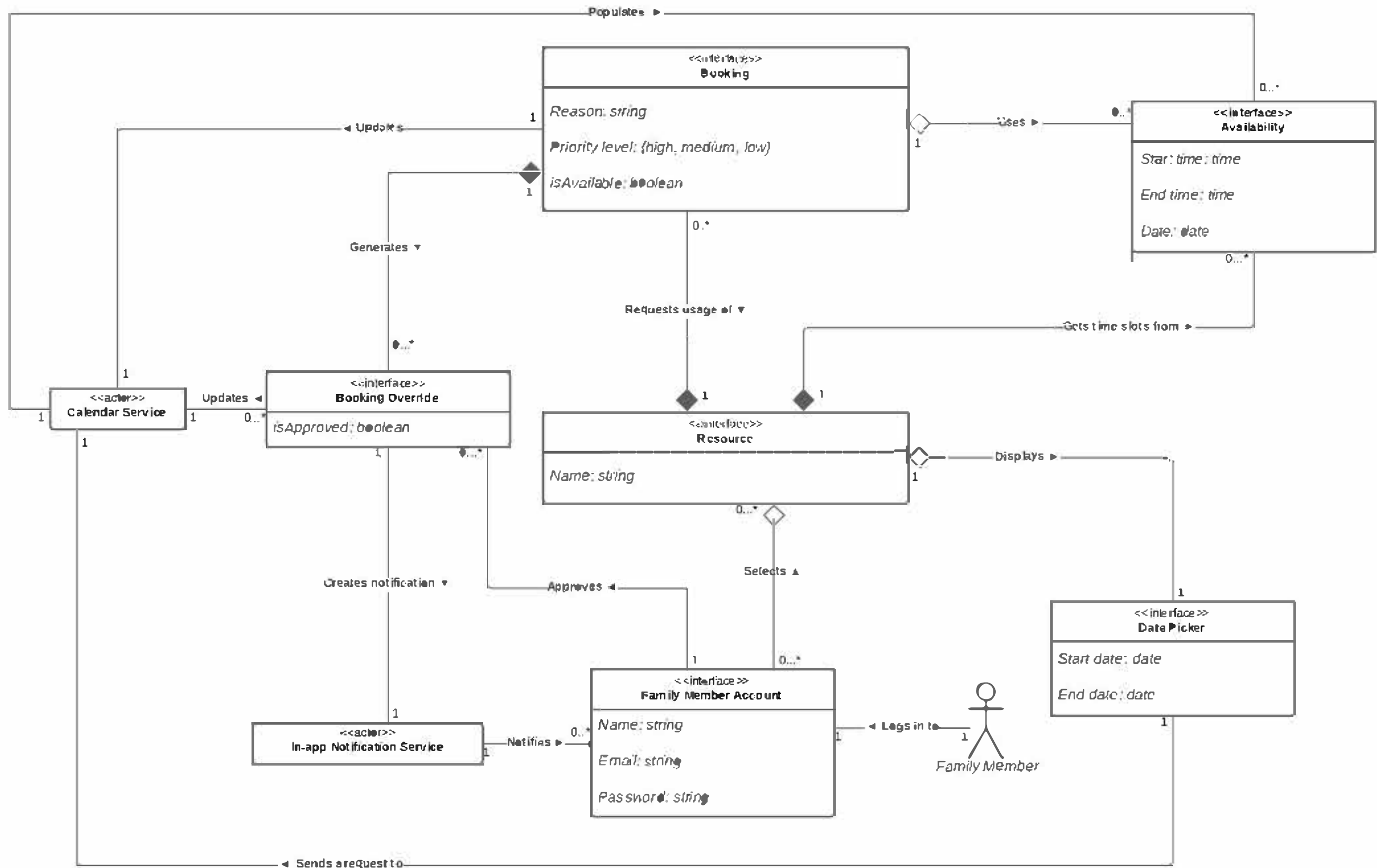


Figure 8: Domain Model of the Coordinate the Usage of Shared Resources Use Case

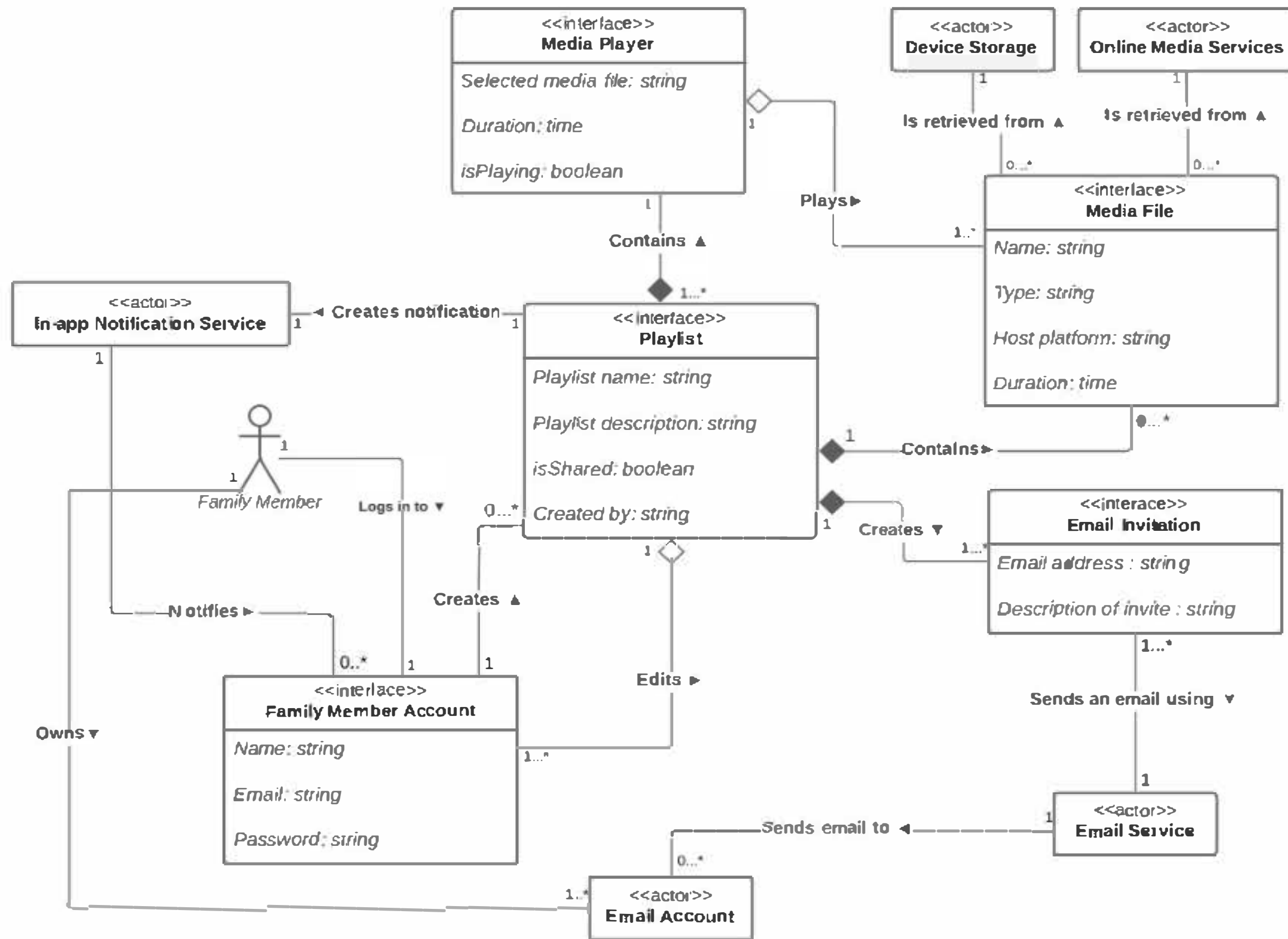


Figure 9: Domain Model of the Collaborate on Shared Media Playlists Use Case

General (Entity Descriptions)

Calendar Service

A service, provided by an adjacent system, that has data about the availability related to each family member account and each shared resource of the family.

In-app Notification Service

A service that constructs and displays corresponding messages depending on the given data.

Availability

An entity that shows an available time slot that can be booked for a resource or a family activity.

Family Member

A human actor who uses the system for family coordination.

Family Member Account

An entity that stores family member information when a family member signs up for the application, providing their name, email address, and password. The account enables family members to go through various workflows on the system.

Use Case 1: Plan a Family Activity (Entity Descriptions)

Family Activity

An interface that allows the family member accounts to host a family activity by scheduling a date and time, inviting other family member accounts, adding a location, and optionally creating a poll to vote for a family activity that the majority of the invitees prefer to attend.

Poll

An interface that includes poll options that each invitee can vote for, with regards to the options the host inputs for potential family activities. It also has a corresponding deadline that specifies the latest time for voting, and the poll is closed when the deadline passes.

Poll Option

An option that is subject to a poll, which contains the name and the description of a family activity, so family member accounts can view the details of a proposed family activity. It also includes the count of the votes for each option, so the system can see which option is the most popular among votes submitted by the invited family members.

Scheduler

An interface that allows the host family member to specify scheduling criteria, including a start date, end date, intended duration of a family activity and the time(s) of day that the activity would occur during. The information in this entity is used to fetch available time slots for all invited family members from the calendar service.

Activity Type

An interface that allows the family member to select a preset type for a family activity or create a custom one.

Task

A responsibility, based on the family activity type, which can be assigned to an individual family member account.

Use Case 2: Coordinate the Usage of Shared Resources (Entity Descriptions)**Booking**

An interface to input the reason and the priority level of using a resource. If the usage time range is not within the fetched availabilities, the booking notes the selected time range as unavailable through the attribute `isAvailable`.

Booking Override

A request that is used when multiple family members want to use the same resource during overlapping time ranges.

Resource

A representation of a shared family resource, such as a car, a monitor, or a house key. The interface simply contains the name of the resource.

Date Picker

An interface for the family member to select a desired date range which narrows down the data returned by the calendar service.

Use Case 3: Collaborate on Shared Media Playlists (Entity Descriptions)**Online Media Services**

An adjacent system that serves as a digital media service (e.g. Spotify, Apple Music, Youtube, etc.) that provides data in the form of media files.

Device Storage

Device storage that contains media files, on a family member's local device, that the family member may want to upload to the playlists.

Media File

An entity that represents media, such as a song, video, picture, etc. It stores the name, the type, the host platform, and the duration of the media file.

Playlist

An interface to create, view, play, and edit a list of media files that family member accounts have added. It supports sharing as well, which enables other family member accounts, who are not the creator, to edit the playlist.

Media Player

An interface to play a media file. All playlists share one media player, and it stores the current media file that has been selected to play.

Email Invitation

An interface to allow the family member to preview an email invitation, and enter the email addresses of family members to share the playlist with.

Email Service

A service that sends data, in the form of email invitations, to the family member accounts when a playlist is created, and a family member account is invited to collaborate on it.

Email Account

An email account is external to the system and is a service that allows family members to send and receive emails. It is assumed every family member owns an email account.

General (Relationship Descriptions)

A Family Member logs in to a Family Member Account

The human actor, which is the family member, is associated with an account in our system. Since the system is only aware of the family member's account, it acts as the representative of the human actor in our modelling. This relationship is one-to-one as there is one account per family member.

The In-App Notification Service notifies the Family Member Account

Our in-app notification service receives a trigger from various other entities in our system and fires the alert to the appropriate family member accounts. These triggers are referenced in all three of our use cases. This relationship is one-to-many since the service can fire alerts to any number of family member accounts.

Use Case 1: Plan a Family Activity (Relationship Descriptions)

A Family Member Account hosts a Family Activity

The family member account that would like to initiate a family activity can choose to host one. This substantiates an instance of a family activity. This relationship is one-to-many as a single account can host as many activities as they please.

A Family Activity invites a Family Member Account

Once the family activity has been finalized, other family member accounts are invited to partake in the activity. Multiple family activities can send invites to multiple family member accounts, hence the relationship is many-to-many. Moreover, many family activities are experienced by one or more family member accounts, hence the relationship is aggregation.

A Family Activity creates a Poll

Once the family member account, that hosts the family activity, creates an instance of the family activity, a poll can be created. A family activity can have 0 or 1 polls created for said activity. Furthermore, this is a composition relationship because each poll is created from an instance of a family activity.

A Poll displays Poll Options

The poll can display multiple poll options. Since a single poll can contain 2 or more poll options (as it would not make sense to create a poll with less than 2 options), this is indicated by the

multiplicity symbols. If a poll is deleted, then its poll options are subsequently deleted, hence the composition relationship.

A Poll is sent through the In-app Notification Service

The family activity poll must be sent to all invitees of the family activity. This is done through the in-app notification service. Because there is only one family activity poll being sent to the notification service, this is a [1, 1] relationship.

A Family Member Account selects Poll Option(s)

A family member, represented by their account, can select their preferred family activity. Each member is allowed to make any number of selections but must select at least one preferred activity from the poll, which is indicated by 0... * and 1... * multiplicity symbols.

A Family Activity selects an Activity Type

Each family activity has one associated activity type. This relationship is one-to-one, as denoted by the multiplicity symbols. Since each family activity can have one family activity type, this is marked by the aggregation relationship.

An Activity Type generates Task(s)

The family activity type listed under the family activity is used to generate 0 or more tasks as indicated by multiplicity symbols. If the family activity type no longer exists, then the task no longer exists as well. This is indicated with the composition relationship. These tasks are used to plan the activity and are delegated to family members.

A Task assigns a Family Member Account

Each task will have a family member account assigned to it by the host. Since any number of tasks can be assigned to one family member account, this shows the many-to-one relationship between these entities.

A Family Activity updates the Calendar Service

Once the family activity is finalized, each participating family member's calendar must be updated with the date, time and other associated details of the activity. A family activity can update multiple family member calendars, however, the service is only pinged once to bulk update these calendars. This is indicated by the one-to-one relationship.

A Family Activity displays a Scheduler

The family activity displays the scheduler interface where family members can specify the criteria for scheduling the activity. This includes selecting a range of desired dates to have the family activity occur on, the intended duration of the activity and the time(s) of day that the family activity should occur during.. Each family activity has only one scheduler interface as indicated by the one-to-one relationship between these entities. Furthermore, each family activity has its own scheduler interface. This is marked by the composition relationship.

A Scheduler sends a request to the Calendar Service

The scheduler narrows down the selection of data that is requested from the calendar service. Since a single scheduler sends a request, based on the date range, intended duration and time(s) of day to the calendar service, this shows the one-to-one relationship between the entities.

The Calendar Service populates an Availability

The calendar service can obtain the various dates and times in which all family members are available, based on the criteria from the scheduler. This populates the availability interface entity. Because the calendar service can populate multiple availabilities based on the scheduling criteria, there is a one-to-many relationship between the calendar service and the availability.

A Family Activity gets a time slot from an Availability

A family activity can only happen during one time slot that is denoted by an availability. Hence, a single availability will be selected for the family activity, thus showing the one-to-one relationship between these entities. As the availability belongs to the family activity, this is marked by the composition relationship.

A Family Activity creates a notification via the In-app Notification Service

Once the family activity has been finalised, a notification is triggered via our in-app notification service. The service then fires the alert to all the associated family member accounts that are invited to the family activity. Since one family activity creates one notification, this is shown through the one-to-one relationship.

The In-app Notification Service notifies a Family Activity

A notification is sent to the family activity once the poll deadline has passed and no family members voted for activities through the poll. In this case, the host family member will receive the notification and, from there, manually enter an activity name and description before proceeding to the scheduling step for the family activity. As one notification gets sent to one family activity, this marks the one-to-one relationship between these entities.

Use Case 2: Coordinate the Usage of Shared Resources (Relationship Descriptions)

A Family Member Account selects a Resource

The family member account can select a resource from the family's inventory of resources. They may wish to select multiple resources to use, and each resource can be selected by multiple family member accounts, so this is a many-to-many relationship. Moreover, many resources are shared by one or more family members, hence the relationship is aggregation.

A Resource gets a time slot from an Availability

A given resource is available only during certain time slots. Hence, it can get a time slot from any of the availabilities. Because a resource can have multiple availabilities, this is denoted by one-to-many multiplicity symbols. Also, if the resource no longer exists, then its associated availabilities are deleted. This is portrayed through the composition relationship.

A Booking requests usage of a Resource

A given resource also has a booking interface. Because a resource can have multiple bookings associated with it, this is marked by one-to-many multiplicity symbols. If the resource no longer exists, its associated bookings are deleted which is indicated by the composition relationship.

A Resource displays a Date Picker

A given resource requires a date picker interface, such that the family member may input the date range for which they would like to book the resource. Furthermore, multiple resources can have the same date picker interface. This is marked by the aggregation relationship.

A Booking updates the Calendar Service

Once a booking is finalised, the resource's calendar must be updated via the calendar service. Since a booking pings only one calendar service, this relationship is one-to-one.

A Booking uses an Availability

A resource booking uses time slots from an availability. Once the booking has been deleted, it frees up its associated availabilities for the resource, so it is an aggregation relationship. Furthermore, for one booking, there may be 0 to many availabilities, hence the one-to-many relationship.

A Booking generates a Booking Override

If a family member would like to override another family member's booking, then a booking override interface is required. Each booking generates a booking override interface. This relationship is one-to-many as there could be multiple overrides per booking. Moreover, once a booking is removed, its booking overrides are also deleted as marked by composition.

A Booking Override Request creates notification via the In-app Notification Service

Once a booking override request is created, a notification is triggered via our in-app notification service to let the original booking owner know that they need to approve the request. This is a one-to-one relationship.

A Family Member Account approves a Booking Override Request

The family member of the original booking can now approve or deny a booking override request. This relationship is one-to-many as there can be multiple override requests that a family member account can approve or deny.

A Booking Override requests updates the Calendar Service

If a booking override has been approved, then it must request the calendar service to update a family member's calendar by cancelling the original booking and updating it with the new one. This relationship is many-to-one as there could be multiple overrides requesting to update via the calendar service.

The Calendar Service populates an Availability

The calendar service can obtain the various dates and times in which the resource is available, based on the date range stored in the date picker. This populates the availability interface entity. Because the calendar service can populate multiple availabilities based on a single date range selection, there is a one-to-many relationship between the calendar service and the availability.

A Date Picker sends a request to Calendar Service

The date picker narrows down the selection of data that is requested from the calendar service. Since a single date picker sends a request, based on the date range, to the calendar service, this shows the one-to-one relationship between the entities.

Use Case 3: Collaborate on Shared Media Playlists (Relationship Descriptions)

A Family Member owns an Email Account

Each family member owns an email account, which is used to send and receive emails. We assumed that every family member has an email that can be used to register for an account. A family member can own multiple email accounts, which is denoted by the 1 to [1...*] relationship.

A Family Member Account creates a Playlist

A family member, represented by their account, can create a new media playlist. Each account can create multiple playlists as denoted by one-to-many multiplicity symbols.

A Family Member Account edits a Playlist

A family member, represented by their account, can edit a media playlist. Moreover, a playlist can be edited by 1 to many family members. The playlist contains many family member accounts that can edit it as represented by the aggregation relationship.

A Playlist creates an Email Invitation

Once a playlist is created, the creator may wish to share the playlist with others. Doing so creates an email invitation entity. This is a one-to-many relationship because a playlist can create as many email invitations as needed. This is also a composition because email invitation is owned by a playlist.

An Email Invitation sends an email using the Email Service

Once the email invitation is created, the email service is pinged to send out the email invitations to collaborate. This is a many-to-one relationship because multiple email invitations can request the email service at the same time.

The Email Service sends an email to an Email Account

The email service completes the issuing of email invitations to an email account, which is owned by a family member. This is a one-to-many relationship as the email service can send the same email invite to multiple email accounts.

A Playlist contains a Media Player

To play the media files within a playlist, it must contain a media player entity. Since there are potentially many playlists and only one media player, this is a [1, 1..*] relationship. Moreover, since a media player cannot be accessed if there are no playlists, this is a composition relationship.

A Media Player plays Media File(s)

The media player entity plays the individual media file(s) found within a playlist. Since there is only one media player for one or more media file(s), this is marked by [1, 1..*] relationship.

A Playlist contains a Media File

A playlist is made up of multiple media files. This relationship is one-to-many as a single playlist typically contains multiple media files. Moreover, if the playlist is deleted then, in the context of our system, the media file is also deleted. Hence, this is denoted by the composition relationship.

A Media File is retrieved from the Device Storage

A media file can be retrieved from local device storage, such as from a family member's phone. The file is then displayed in a playlist. This relationship is many-to-one as multiple media files can be obtained from device storage.

A Media File is retrieved from Online Media Services

A media file can also be retrieved from an online media service (e.g., the cloud). The file is then displayed in a playlist. This relationship is many-to-one as multiple media files can be obtained from the online media service.

A Playlist creates a notification via the In-app Notification Service

Once the media playlist has been created and shared with family members, a notification about how a playlist is shared with them is triggered via our in-app notification service. The service then fires the alert. This is a one-to-one relationship between the playlist and the in-app notification service.