Компания «Мир загадок» занимается проведением телевизионной игры «Загадки». Используя метод мозгового штурма, команда игроков ищет в течение двух минут правильный ответ на загадку телезрителя. За правильный ответ очко получает команда игроков, за неправильный же очко получают их противники — команда телезрителей. Побеждает команда, первой набравшая семь очков.

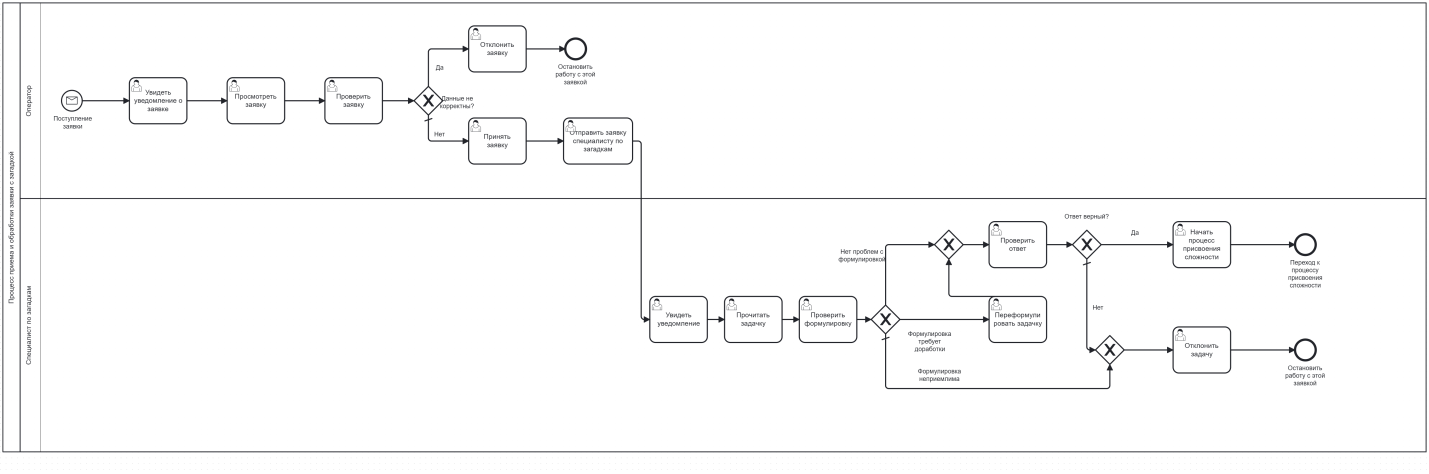
Загадки, которые присылают телезрители, могут быть простые, средней сложности и сложные. Всего в игре участвует 13 загадок, из них 4 простого, 4 среднего, 4 сложного и 1 неизвестного уровня сложности (может быть простой, средней или сложной). Уровень сложности загадки определяет специалист по загадкам, когда они поступают от телезрителей. Загадки хранятся в базе загадок.

Игра проходит каждую неделю по пятницам в прямом эфире на телеканале Загадочный.

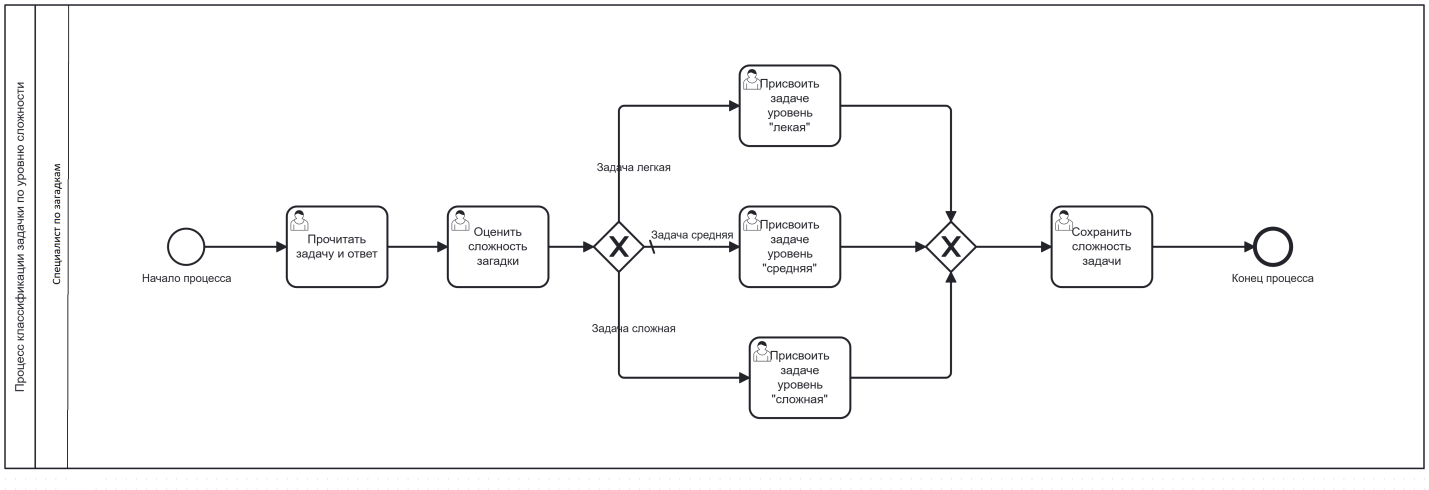
3.1. Верхнеуровневое моделирование системы, основными задачами которой является прием, обработка заявок с загадками от телезрителей, классификация загадок по уровню сложности, отбор загадок на игру и оповещение телезрителей-авторов загадок, которые были разыграны в прямом эфире, об их выигрыше или проигрыше.

Модель процессов

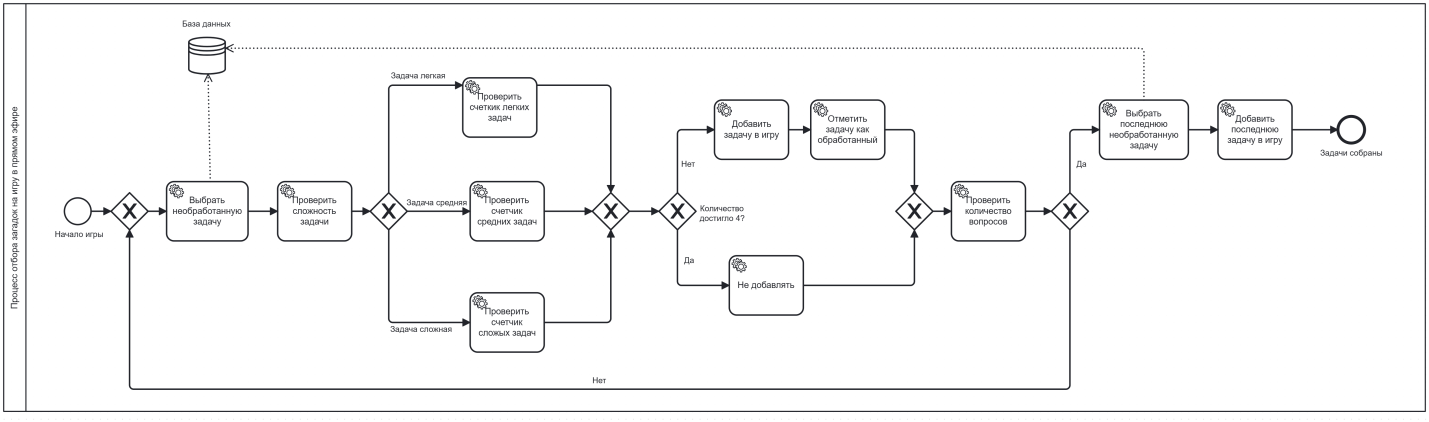
Процесс приема и обработки заявки с загадкой.



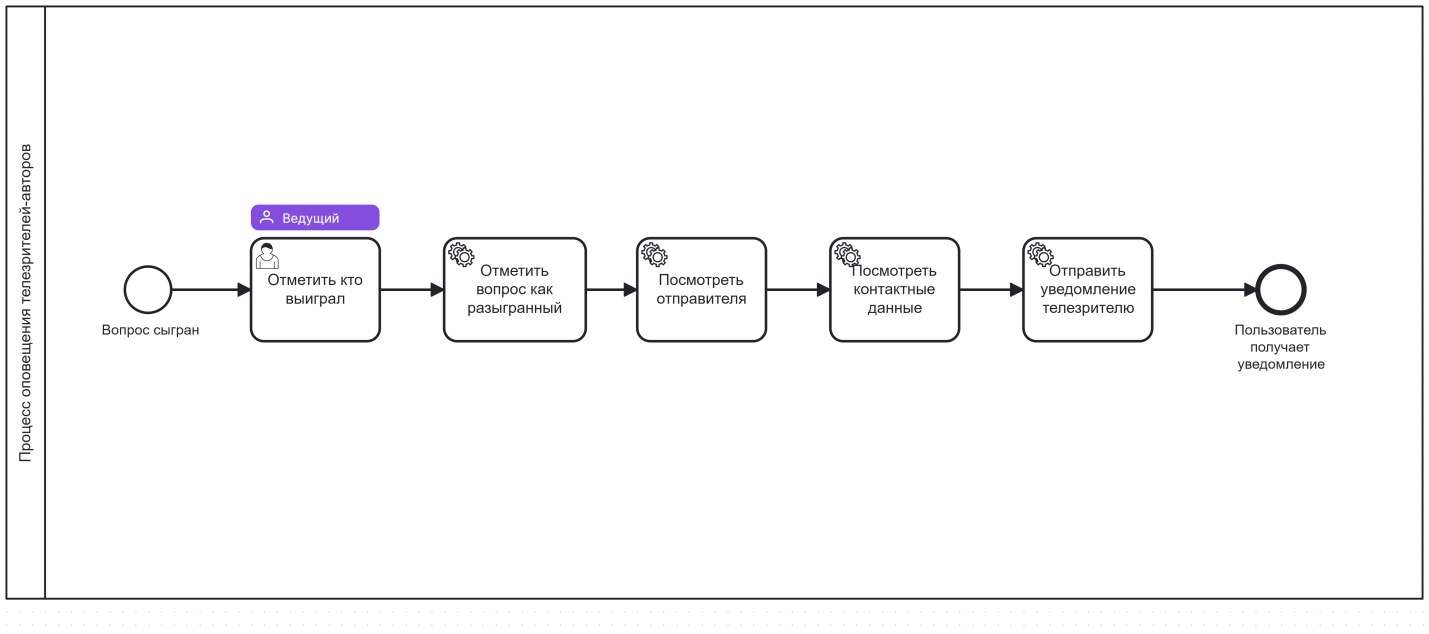
Процесс классификации загадки по уровню сложности.



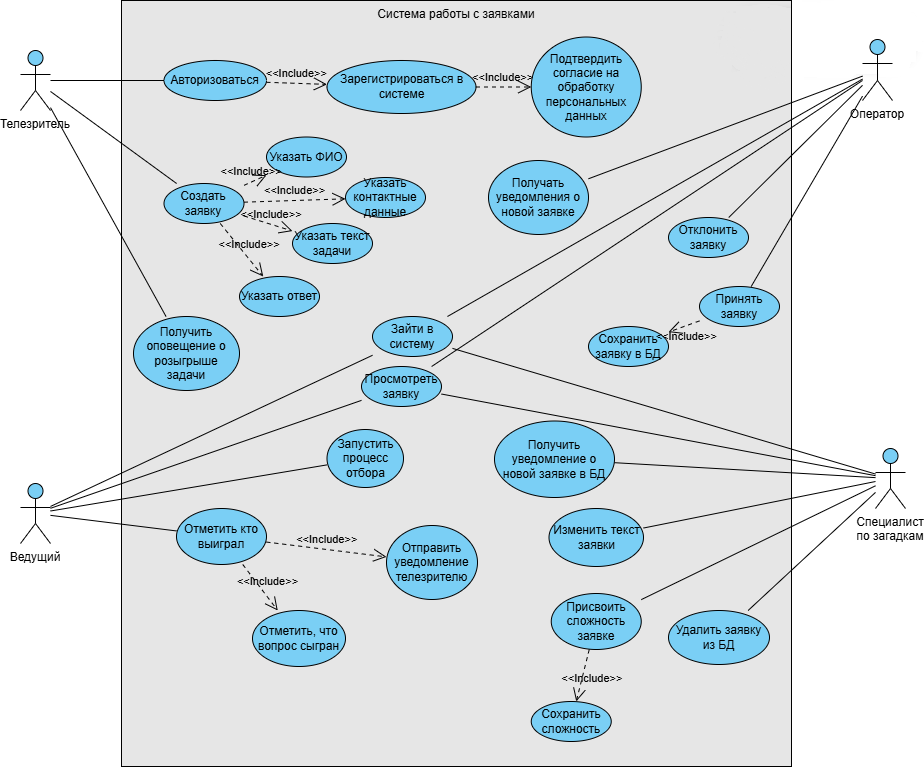
Процесс отбора загадок на игру в прямом эфире.



Процесс оповещения телезрителей-авторов загадок, которые были разыграны.



Функциональная модель



UseCase диаграмма функционала системы.

|  |  |
| --- | --- |
| Пользовательское требование | Цель |
| Требования телезрителя | |
| Авторизоваться | Идентифицировать телезрителя |
| Зарегистрироваться в системе | Идентифицировать телезрителя |
| Подтвердить согласие на обработку персональных данных | Получить согласие на просмотр и использование личных данных телезрителя |
| Создать заявку | Получить от телезрителя загадку для БД |
| Указать ФИО | Идентифицировать телезрителя |
| Указать контактные данные | Получить возможность связаться с телезрителем |
| Указать текст загадки | Получить готовый вопрос |
| Указать ответ | Получить готовый ответ |
| Получить оповещение о розыгрыше загадки | Проинформировать телезрителя и том, что его загадка была разыграна и о результате |
| Требования ведущего | |
| Зайти в систему | Идентифицировать сотрудника |
| Просмотреть заявку | Прочитать текст и увидеть ответ на загадку |
| Запустить процесс отбора загадок | Собрать нужное количество загадок на игру |
| Отметить, кто выиграл | Зафиксировать результат игры |
| Отправить уведомление телезрителю | Проинформировать телезрителя и том, что его загадка была разыграна и о результате |
| Отметить, что вопрос разыгран | Зафиксировать, что вопрос был использован |
| Требования оператора | |
| Зайти в систему | Идентифицировать сотрудника |
| Просмотреть заявку | Иметь возможность проверить данные телезрителя для его идентификации |
| Получить уведомление о новой заявке | Заметить новую загадку |
| Отклонить заявку | Следить за правильностью контактной информацией |
| Принять заявку | Зафиксировать новую заявку |
| Сохранить заявку в БД | Зафиксировать новую заявку |
| Требования специалиста по загадкам | |
| Зайти в систему | Идентифицировать сотрудника |
| Просмотреть заявку | Иметь возможность проверить текст и ответ загадки |
| Получить уведомление о новой заявке в БД | Заметить новую загадку |
| Изменить текст задачи | Редактировать неточности в формулировке |
| Присвоить сложность заявке | Указать уровень загадки |
| Сохранить сложность | Зафиксировать уровень загадки |
| Удалить задачу из БД | Следить за читабельностью загадок и наличием правильного ответа |

Логическая модель данных

Сущность «Телезритель»

Атрибуты:

* код\_телезрителя: int
* ФИО: string
* телефон: number
* email: string

Методы:

* получить\_код(): void
* получить\_уведомление(): void

Сущность «Сотрудник»

Атрибуты:

* код\_сотрудника: int
* ФИО: string
* телефон: number
* email: string
* паспорт: int
* снилс: int
* должность: string
* договор: int

Методы:

* отправить\_код(param: number): boolean
* отправить\_уведомление(param: number): boolean
* прочитать\_заявку(param: int):void
* сохранить\_заявку(param: int):void
* изменить\_заявку(param: int):void
* удалить\_заявку(param: int):void

Сущность «Заявка»

Атрибуты:

* код\_заявки: int
* код\_телезрителя: int
* код\_проверевщего\_сотруника: int
* код\_утвердившего\_сотрудника: int
* текст\_загадки: string
* ответ: string
* сложность: string
* отправитель\_победил: boolean
* заявка\_обработана: boolean

Методы:

* подтвердить\_код(param: number): boolean
* показать\_заявку(): void

Сущность «Сборник»

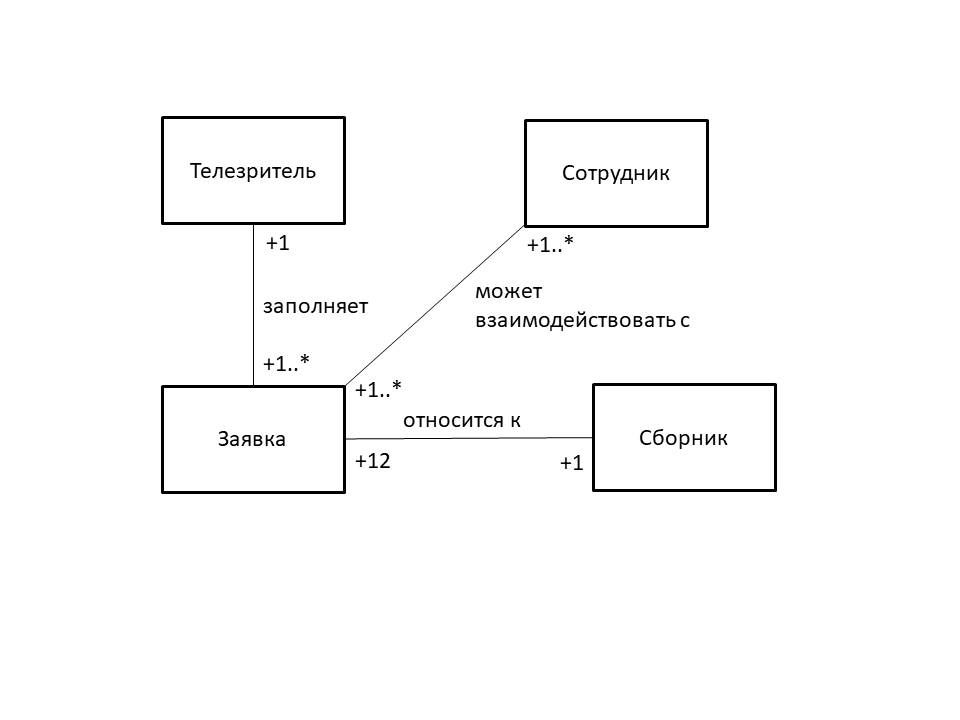
Атрибуты:

* код\_сборника: int
* код\_заявки1: int
* код\_заявки2: int
* код\_заявки3: int
* код\_заявки4: int
* код\_заявки5: int
* код\_заявки6: int
* код\_заявки7: int
* код\_заявки8: int
* код\_заявки9: int
* код\_заявки10: int
* код\_заявки11: int
* код\_заявки12: int
* код\_заявки13: int

Методы:

* показать\_сборник(param: int): void

Отношения между основными сущностями показаны на диаграмме:



3.2. Нефункциональные требования

1. При регистрации телезритель не может нажать на кнопку «Зарегистрироваться», если не подтвердил согласие на обработку персональных данных.
2. При регистрации телезритель подтверждает свои контактные данные кодом из сообщения.
3. При взаимодействии с системой время ответа системы не больше 5 секунд.
4. Резервное копирование данных должно выполняться раз в сутки.

3.3. Описание алгоритма, по которому система выбирает загадки для игры.

1. В системе храниться три счетчика, равные 4. И количество уже отобранных заявок, равное 0.
2. Система берет из БД заявку, не отмеченную обработанной.
3. Система проверяет сложность задачи.
4. Если счетчик этой сложности > 0, то задача добавляется в игру, а задача помечается обработанной.
5. Процесс (п.2-4) повторяется, пока количество отобранных заявок < 13.
6. После чего система берет из БД заявку, не отмеченную обработанной, и добавляет ее в игру.

Алгоритм может быть также адаптирован при возможности использования SQL.

1. Производиться запрос на 4 задачи легкой сложности из необработанных задач.
2. Производится запрос на 4 задачи средней сложности из необработанных задач.
3. Производиться запрос на 4 задачи сложной сложности из необработанных задач.
4. Производится запрос на любую одну необработанную задачу.