Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

Техническое задание

По теме «Платформа для размещения и проведения курсов»

Выполнила:

Студентка гр. 953502

Мазец М.С.

Руководители:

Гриценко Никита Юрьевич

Тушинская Екатерина Вадимовна

Минск 2022

# **Введение**

В данное время образование в нашей жизни играет ключевую роль. Все хотят иметь хорошую работу, быть интересным собеседником, разбираться в литературе или искусстве, моде или здоровом питании, политике или экономике. События последних лет показали, что обучение «на месте», не всегда является доступным, быть может из-за очередной волны вируса или войны. Тем не менее учиться хочется и надо, поэтому было бы хорошо иметь платформу, где можно найти интересующую нас тему, записаться на курс и получать знания с любой точки мира. Таким образом, целью проекта является разработка прогрессивного веб-приложения, которое сочетало бы в себе все функции и процессы, необходимые для организации и проведения развивающих курсов: возможность пользоваться предоставленным материалом, отслеживать свой прогресс, а также оценивать курсы. В процессе разработки будут задействованы современные инструменты для разработки веб-приложения.

# **Требования к проектируемому программному средству**

1. ***Назначение проекта***

Назначением проекта является разработка программного средства, обеспечивающего размещение и проведение образовательных курсов.

1. ***Основные функции***

Программное средство должно поддерживать следующие основные функции:

* регистрация, аутентификация, авторизация;
* поддержка системы ролей;
* управление профилями пользователей;
* категоризация представленных курсов;
* управление профилями курсов;
* управление подписками на предоставленные курсы;
* категоризация материалов курсов.

1. ***Дополнительные функции***

После анализа существующих аналогов, а также определения основных функций приложения, было принято решение о выдвижении дополнительных функциональных требований к проектируемому программному средству:

* возможность оплаты курсов;
* возможность оценивания курсов.
* начисление баллов за прохождение курсов.

1. ***Требования к входным данным***

Входные данные для программного средства должны быть представлены в виде вводимого пользователем с клавиатуры текста, документов соответствующего формата и содержания и опций, предоставляемых пользовательским интерфейсом приложения. В бизнес-логике программного средства должны быть реализованы проверки входных данных на корректность, и в случае их невалидности, пользователь должен получать соответствующее уведомление с просьбой ввести данные необходимого формата.

1. ***Требования к выходным данным***

Выходные данные должны быть представлены посредством отображения информации при помощи различных элементов реализованного и доступного пользовательского интерфейса.

1. ***Требования к аппаратному обеспечению клиентской части***

Клиентская часть программного средства должна функционировать на ЭВМ со следующими минимальными характеристиками:

* процессор Intel Сore с тактовой частотой 2 ГГц и более;
* оперативная память 1 Гб и более;
* возможность выхода в сеть Интернет;

Для корректной работы программного средства необходим один из следующих браузеров с соответствующей минимальной версией:

* Google Chrome 70;
* Opera 58;
* Mozilla Firefox 66;
* Microsoft Edge 44.

1. ***Выбор инструментов разработки***

На основании выдвинутых к программному средству основных и дополнительных функциональных требований, было принято решение о проектировании кроссплатформенного программного средства, функционирующего в сети Интернет.

**Язык программирования C#**

**C#** — объектно-ориентированный язык программирования, который используется в том числе и для разработки веб-приложений. Язык ориентирован на повышение производительности разработчика и читаемости кода.

 Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

**ASP.NET Core MVC**

ASP.NET Core MVC представляет собой упрощенную, эффективно тестируемую платформу с открытым исходным кодом, оптимизированную для использования с ASP.NET Core.

ASP.NET Core MVC предоставляет основанный на шаблонах способ создания динамических веб-сайтов с четким разделением задач. Она обеспечивает полный контроль разметки, поддерживает согласованную с TDD разработку и использует новейшие веб-стандарты.

Структура архитектуры MVC разделяет приложение на три основных группы компонентов: модели, представлении и контроллеры. Это позволяет реализовать принципы разделения задач. Согласно этой структуре запросы пользователей направляются в контроллер, который отвечает за работу с моделью для выполнения действий пользователей и получение результатов запросов. Контроллер выбирает представление для отображения пользователю со всеми необходимыми данными модели.

На следующей схеме показаны три основных компонента и существующие между ними связи.



Рисунок 1. Связь между компонентами MVC

Для хранилища данных выбран MSSQL Server, которым легко пользоваться при разработке приложений на .NET. Для работы с базой данных ASP .NET Core использует EntityFramework Core. Таким образом, можно описать модель данных с помощью классов C#, после чего схема БД будет сгенерирована автоматически. Кроме того, данный фреймворк имеет удобный функционал для манипуляции над данными.

# **Разработка функциональных требований**

## **Функциональная модель программного средства**

Исходя из выдвинутых требований, проектируемое программное средство предполагает разделение пользователей на различные роли и выдача различных прав нескольким категориям пользователей: *гостю, студенту, преподавателю.*

Для гостя доступны регистрация, аутентификация и авторизация.

После регистрации пользователь получает роль – «Студент» или «Преподаватель». «Студент» имеет доступ к функциям управления профилем, функциям управления курсами, на которые он подписан.

*Функции управления профилем включают в себя:*

* Редактирование личной информации. Сюда входят возможности заполнения и редактирования имени, фамилии, адреса электронной почты, мобильного телефона, загрузки фотографии профиля, привязка карточки для оплаты.
* Просмотр информации о начисленных баллах за прохождение различного вида платных курсов.
* Оформление подписки.

*Функции управления курсом:*

* + Просмотр материалов курса. Сюда входят возможность просмотров всей информации, которую предоставляет преподаватель курса, на который студент оформил подписку, доступ к материалам.
  + Просмотр оставленных оценок другими пользователями.
  + Взаимодействие с преподавателем.

*Функции управления профилем для пользователя «Преподаватель» включают в себя:*

* + Редактирование личной информации. Сюда входят возможности заполнения имени, фамилии, адреса электронной почты, мобильного телефона, загрузки фотографии профиля, привязка карточки.

Функции управления курсом для пользователя «Преподаватель»:

* + Добавление/редактирование материалов курса. Сюда входят возможность добавление новой информации и редактирование уже представленной для курсов, которые закреплены за пользователем.
  + Взаимодействие с группой. Данная функция предоставляет возможность общаться с группой учащихся, подписанных на курс.

Mind card разрабатываемого приложения (рисунок 2):

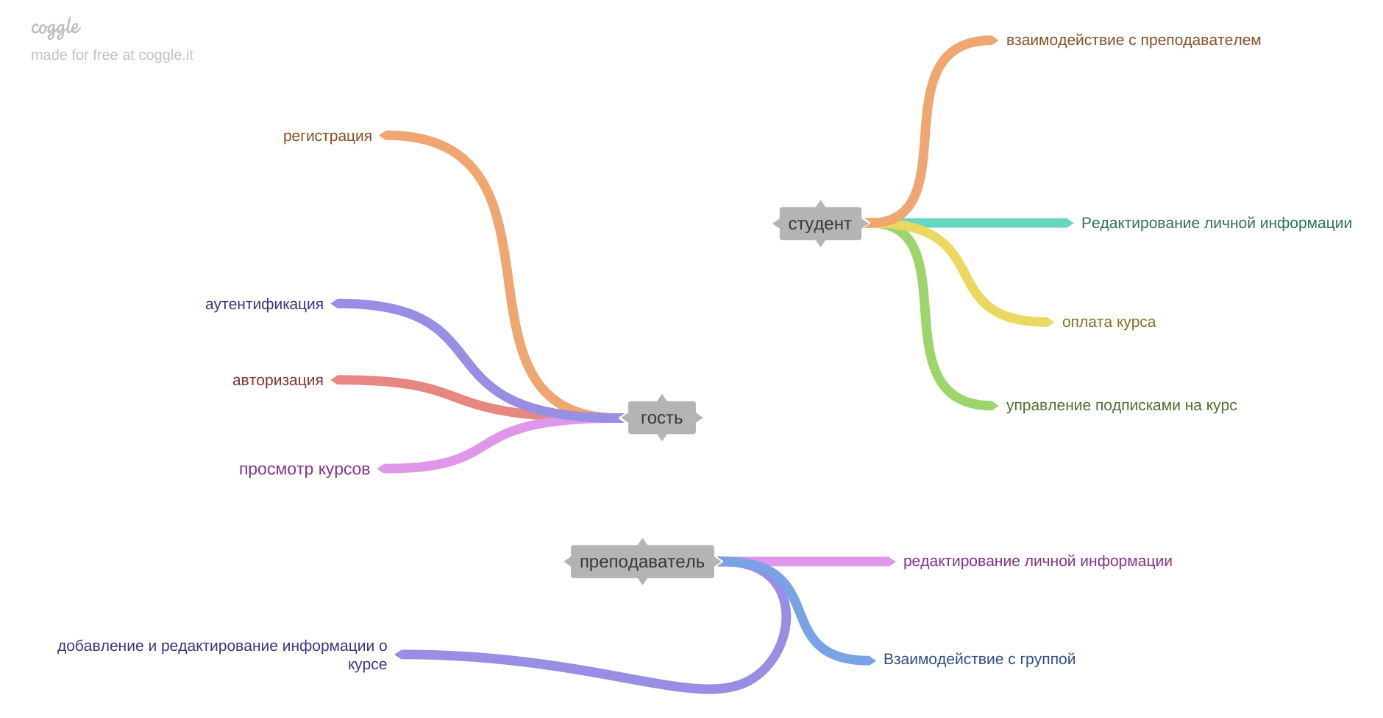


Рисунок 2. Mind card разрабатываемого приложения

## **Разработка модели базы данных**

Предметная область разрабатываемого программного средства включает в себя следующие сущности и их сущности:

Пользователь:

* ник пользователя
* хэшированный пароль
* электронная почта;
* номер телефона
* роль
* список подписок
* список начисленных за прохождения курсов баллов
* уникальный идентификатор карточки;

Роль:

* название роли
* пользователи: коллекция объектов, через которые пользователи ассоциированы с данной ролью

Кредитная карточка:

* хэшированный номер карточки
* срок действия

Курс:

* название
* дата начала
* дата окончания
* преподаватель
* короткое описание
* стоимость (может быть NULL)
* валюта (может быть NULL, когда курс бесплатный)
* уровень подготовки
* ссылка на хранилище
* список студентов
* чат

Рейтинг (производиться анонимно):

* уникальный идентификатор курса
* список оставленных оценок

Отзывы:

* уникальный идентификатор курса
* список оставленных отзывов

Хранилище:

* имя хранилища
* имя файла
* дата последнего обновления

Чат:

* является ли чат только для взаимодействия с преподавателем
* список сообщений

Сообщение:

* отправитель
* дата отправки
* текст

Валюта:

* наименование
* доля в сравнении с долларом

## **Требование к внешним интерфейсам**

При входе на главную страницу пользователь видит каталог с курсами (рисунок 3):

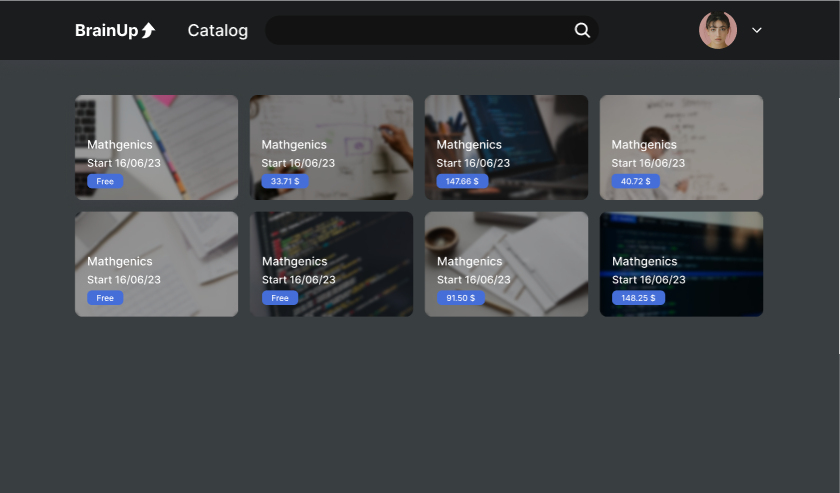


Рисунок 3. Стартовая страница с курсами

Также для фильтрации пользователь может выбрать нужные параметры (рисунок 4):

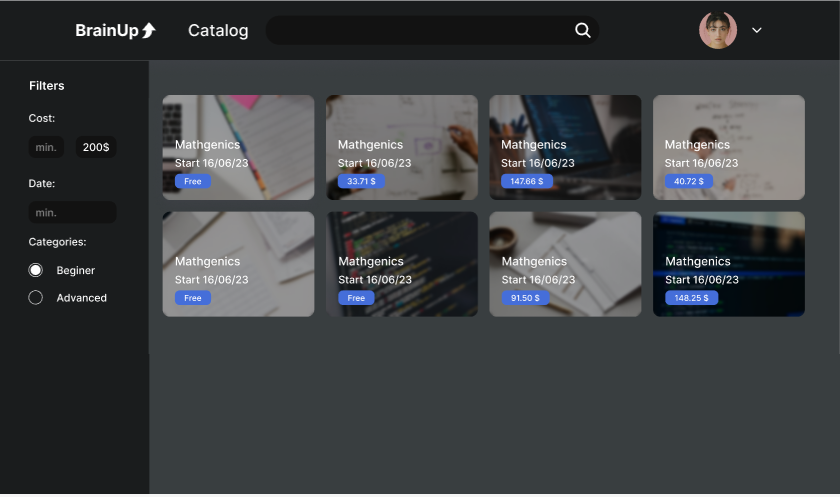


Рисунок 4. Фильтрация курсов

Страница регистрации/авторизации (рисунок 5):

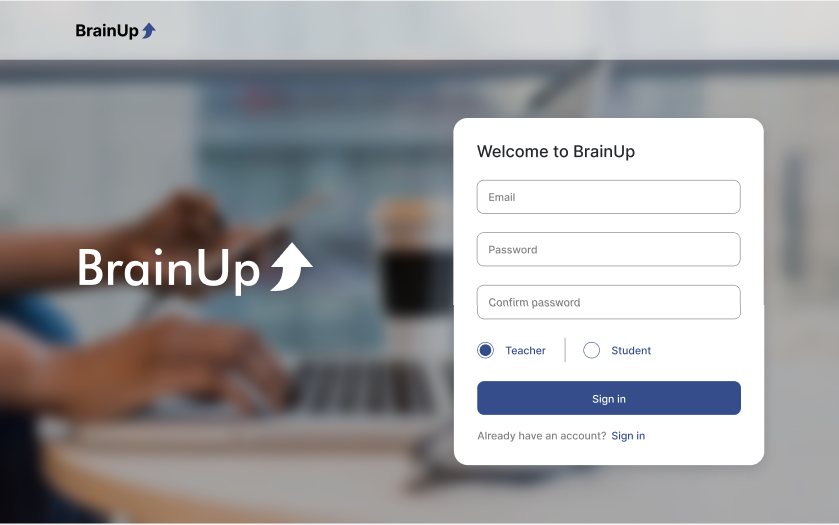
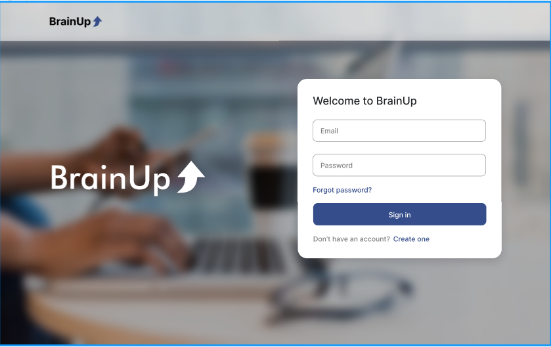


Рисунок 4. Страница регистрации\авторизации

## **Разработка спецификации функциональных требований**

Для детализации функций рассмотрим основные требования, предъявляемые к каждой функции программного средства как с точки зрения внутренней организации системы, так и с точки зрения взаимодействия системы с пользователем.

*Функция регистрации*:

* процесс регистрации инициируется пользователем системы;
* правильность введенного пароля должна быть проверена при помощи встроенных инструментов разработки;
* пользователь должен иметь возможность выбрать в какой роли он хочет выступать: преподаватель или студент.
* в случае некорректности введенных данных пользователь должен увидеть сообщение об этом с предложением попробовать еще раз;
* должна быть предусмотрена возможность смены пароля после регистрации.

*Функция аутентификации:*

* инициатором является пользователь, при этом ему необходимо предоставить имя пользователя и пароль, заданные при регистрации;
* должна быть реализована возможность восстановления пароля:
* для восстановления пароля пользователь должен предоставить мобильный телефон, заданный при заполнении личного кабинета;
* на предоставленный адрес электронной почты присылается новый пароль;
* после входа по данному паролю пользователь может вновь сменить его на желаемый.

*Функция каталогизирования курсов:*

* право на создание курса имеет преподаватель;
* редактирование курса должно производится не позднее даты начала;
* удаление курса должно производится не ранее даты завершения;
* сортировка каталога курсов производится по следующим параметрам: категория, дата начала, уровень, стоимость;

*Функция оплаты курсов:*

* студент может изначально привязать карту либо ввести данные, когда захочет оплатить курс;
* для обеспечения оплаты пользователь должен ввести номер телефона (если его изначально нет в профиле), либо подтвердить существующий. На данный номер будет выслан код, который нужно будет ввести в поле для подтверждения оплаты.
* сумма должна быть списана со счета студента и зачислена на счет преподавателя курса.

*Функция начисления баллов за пройденный платный курс:*

* при покупке платного курса пользователю начисляются баллы;
* баллами можно частично или полностью оплатить курс.

*Функция размещения курсов:*

* любой пользователь, который имеет роль «Преподаватель» может разместить курс;
* обязательно должны быть указаны название, дата начала и окончания;
* если стоимость курса не указана, он является бесплатным и соответственно нельзя выбрать валюту.
* Можно добавить ссылку на хранилище с размещенными материалами.