

GOOSLE

УТВЕРЖДАЮ

Горожанкин В.О. (Team Lead/Архитектор, GOOSLE)

Личная подпись Расшифровка подписи

Печать

Дата

УТВЕРЖДАЮ

Малынковский О.В. (Аналитик - разработчик, GOOSLE)

Личная подпись Расшифровка подписи

Печать

Дата

создание приложения

Контроль заболеваемости COVID в университете

ТРЕБОВАНИЯ

На 6 листах

Действует с «01» апреля 2021 г.

ИСПОЛНИТЕЛИ

Организация	Должность	ФИО	Подпись	Дата
GOOSLE	Team Lead/Архитектор	Горожанкин В.О.		
GOOSLE	Аналитик - разработчик	Малынковский О.В.		
GOOSLE	Разработчик (Back, DB)	Кунафин И.А.		
GOOSLE	Разработчик (Back)	Хабибулин М.И.		

Москва, 2021

Протокол изменений

Номер	Дата	Автор(-ы) изменений	Изменения
1.	04.03.2021	Горожанкин В.О. Слободянский П.А.	- создан документ - добавлено оглавление - сформулированы ФТ для «Студента»
2.	25.03.2021	Горожанкин В.О.	- сформулированы требования: общие, «Преподавателя», «Ректората/Деканата» - добавлены страницы: протокол изменений, оглавление
3.	28.03.2021	Горожанкин В.О. Малынковский О.В.	- сформулированы цели и задачи - сформулированы нефункциональные требования
4.	03.04.2021	Горожанкин В.О.	- формирование общего требования для просмотра расписания вместо отдельного для каждой роли
5.	05.04.2021	Горожанкин В.О.	- корректировка целей и задач
6.	12.04.2021	Горожанкин В.О.	- функциональные и нефункциональные требования детализированы
7.	20.04.2021	Горожанкин В.О.	- добавлены лингвистические требования - добавлены требования к серверу
8.			
9.			
10.			
11.			
12.			

Оглавление

Цель и задачи.....	4
Функциональные требования	5
Общие требования	5
Приложение «Студент»	7
Приложение «Преподаватель»	8
Приложение «Ректорат/деканат»	9
Нефункциональные требования	11

Цель и задачи

Цель – отслеживание уровня заболеваемости вирусом COVID-19 среди студентов.

Задачи:

1. Ведение учета посещения студентов;
2. определение потенциально зараженных студентов на основании посещения;
3. отметка о заражении или здоровом состоянии путем прикрепления подтверждающего документа;
4. демонстрация информации о зараженных студентах для преподавателя;
5. рекомендация и закрытие университета или отдельной кафедры в случае достижения определенного уровня заболеваемости.

Функциональные требования

Общие требования

Номер	Название	Условия/ограничения	Комментарий
1.	Регистрация	Система должна давать возможность каждому пользователю зарегистрироваться в системе (при наличии пропуска). Для регистрации студенту необходимо ввести ФИО, номер студенческого билета, паспортные данные, дату рождения и желаемый пароль. После ввода данные будут отправлены в БД для создания новой записи о пользователе. Ролирование автоматическое в зависимости от первой буквы пропуска: - s - Студент - t - Преподаватель - a - Администрация	
2.	Вход в систему	Система должна давать возможность войти в систему при наличии учетной записи данного пользователя. Для входа пользователю необходимо ввести имя пользователя (номер студенческого) и пароль, после чего система отправит запрос в БД для проверки существования данного пользователя в системе: при положительном результате осуществит вход в систему под указанным пользователем, при отрицательном будет выведена ошибка о некорректности введенных данных.	
3.	Выход из системы	Система должна давать возможность выйти из системы при необходимости. Для этого пользователю будет необходимо нажать на кнопку справа сверху с соответствующим значком.	
4.	Просмотр профиля	Система должна давать пользователю возможность	

		<p>просматривать собственный профиль.</p> <p>На странице профиля человека должна быть отображена информация: ФИО, группа (если студент), дата рождения, контактные данные (телефон, email), студенческий билет, если студент болен – отображаются его дата заболевания и дата выздоровления, если выздоровел. Также на данном экране присутствует кнопка, которая отмечает о болезни.</p>	
5.	Просмотр статистики заболеваемости Covid	<p>Система должна демонстрировать статистику заболеваемости группы и всех учащихся в виде столбчатой диаграммы.</p> <p>На странице просмотра статистики будет отображена информация об общем количестве зараженных по всему университету с приведенным графиком, на котором по вертикали отображал шкала количества зараженных, по горизонтали – дата, когда был заражен студент. Также будет присутствовать фильтр со следующими параметрами – группа, факультет, университет – для того, чтобы пользователь мог указать, в каком разрезе ему необходимо посмотреть информацию.</p>	Ежедневная заболеваемость
6.	Просмотр расписания	<p>Система должна иметь возможность отображать расписание преподавателя и студентов.</p> <p>В системе будет создана страница, на которой будет отображено 6 таблиц, которые будут содержать информацию о расписании для студента, под данными которого был осуществлен вход в систему. В таблице будет информация о занятиях, аудитории, в котором проводится занятие,</p>	

		преподавателе, который будет проводить занятие, а также ссылка на онлайн-конференцию, если занятие проводится в Teams.	
--	--	--	--

Требования к роли «Студент»

Номер	Название	Условия/ограничения	Комментарий
1.	Подключение к сервисам онлайн-обучения	Система должна давать возможность перехода студентами по ссылке к онлайн-занятиям из расписания. Так как в таблице с расписанием присутствует ссылка на онлайн-конференцию, в которой будет проведено занятие, будет реализован переход по ссылке из приложения в браузер для открытия конференции.	
2.	Просмотр списка студентов группы	Студент имеет возможность просматривать свою группу для ознакомления с составом. Для этого будет создана отдельная страница, на которой будет выведен номер группы, количество студентов в ней, а также куратор группы при его наличии. Будет создана таблица, в которой будет отображена в виде столбцов следующая информация: ФИО студента, дата рождения. Также при нажатии на студента будет открываться всплывающее окно, в котором будет представлена более подробная информация по выбранному студенту: его телефон и email.	
3.	Отметка о болезни	Система должна давать возможность отобразить статус своего Covid-состояния. На экране профиля студента будет реализована кнопка, при нажатии на которую будет открываться всплывающее окно, в котором необходимо указать статус своего	Нажатие на кнопку “болен/не болен” Загрузка скана анализов

		заболевания. Для каждого из статусов необходимо будет загрузить подтверждающий данный статус документ, после загрузки которого система оправит данные в БД для подтверждения и изменения статуса студента.	
4.	Получение уведомления в личном кабинете	Система должна отображать в ЛК риск заражения Covid и рекомендацию к обследованию. Если в группе какой-либо студент заболел, то для всех студентов, которые посещали с заболевшим пары на протяжении последней недели, придет уведомление о необходимости пройти проверку на заражение и прикрепление скана о полученных результатах в систему по механизму, описанному в предыдущем шаге.	

Требования к роли «Преподаватель»

Номер	Название	Условия/ограничения	Комментарий
1.	Простановка посещаемости	Система должна давать возможность преподавателю проставлять присутствие студента на занятии в определенную дату. Для преподавателя будет реализована отдельная страница с занятием. На нем преподавателю будет предложено выбрать группу и дату занятия, после чего система отправит запрос в БД с указанными фильтрами и выведет в таблицу информацию по найденному занятию по указанным данным. В таблице будет отображена информация: ФИО студента, а также будет возможность проставлять баллы и посещаемость студента. После сохранения данных путем	

		нажатия на кнопку, запрос на изменение БД будет отправлен.	
2.	Простановка баллов	Система должна давать возможность преподавателю проставлять целочисленные баллы каждому из студентов, кто присутствовал на занятии. В таблице будет отображена информация: ФИО студента, а также будет возможность проставлять баллы и посещаемость студента. После сохранения данных путем нажатия на кнопку, запрос на изменение БД будет отправлен.	
3.	Просмотр заболеваемости	Система должна иметь возможность отображать статус заболевания студента в общем списке для преподавателя. В таблице, в которой преподаватель просматривает информацию о ФИО студента и проставляет его баллы за занятие и посещаемость, цветовой палитрой красного оттенка будут выделены студенты, которые сейчас больны и не могут физически посещать занятия, чтобы не заражать сокурсников.	

Требования к роли «Ректорат/деканат»

Номер	Название	Условия/ограничения	Комментарий
1.	Рекомендация к закрытию университета/кафедры	Система должна иметь возможность выводить уведомление с рекомендацией о закрытии всего университета или кафедры. При достижении уровня заболеваемости свыше 60% система будет выводить для данной роли соответствующее сообщение, которое будет рекомендовать кафедру, либо университет к закрытию, после чего открывается возможность нажатия кнопки, которая осуществляем	

		функционал «закрытия» кафедры/университета.	
2.	Закрытие университета/кафедры	<p>Система должна иметь возможность закрытия университета/кафедры и уведомление всех причастных к данному вузу/кафедре студентов о выполненной мере.</p> <p>Будет реализовано посредством нажатия на кнопку, после чего для соответствующего университета или кафедры (смотря что закрывают) проставится соответствующий статус, после чего студенты данной группы будут переведены на дистанционную форму обучения.</p>	

Нефункциональные требования

Номер	Название	Условия/ограничения	Комментарий
1.	Производительность	Время отклика системы должно не превышать 4 секунд. Количество пользователей, который должны иметь одновременный доступ к системе – 15.000.	
2.	Масштабируемость	Должна быть возможность как вертикального, так и горизонтального масштабирования серверов. Вертикальная масштабируемость необходима для увеличения производительности сервера, на котором находится наше приложение путем добавления дополнительных ОЗУ или замены компонентов на обновленные версии. Необходима для предупреждения ситуации, когда мощности приложения будет не хватать в связи с растущим количеством пользователей, одновременно использующих приложение. Горизонтальная масштабируемость – добавление серверов для увеличения места для хранения информации при заполнении уже текущих дисков, задействованных для функционирования системы.	
3.	Безопасность	Серверная часть системы должна быть расположена на собственных серверах (on-premise) по ряду причин: - потенциальная кража данных, если приложение находится в облачном окружении другого производителя, взлом данной платформы может предоставить доступ также к нашим данным; - стабильность – нарушение работы облачной платформы может повредить возможность функционирования и нашего	

		приложения, в связи с чем может уменьшиться спрос. Защита учетных записей пользователей производится за счет пароля, указываемого при регистрации.	
4.	Удобство использования	Интерфейс пользователя должен быть интуитивно понятным. Все элементы в системе должны быть реализованы по принципу «важная информация находится в слева сверху, менее справа снизу». Слева будет реализована статичная панель навигации, в которой будут отображены страницы, которые доступны пользователю для посещения, верхняя часть с информацией об университете, группе, а также студенте, который на текущий момент авторизован в системе также будет статична.	
5.	Используемые технологии	Windows Forms (C#), в качестве БД - MySQL (AWS), для обращения к БД будет использован Entity Framework.	
6.	Лингвистические требования (поддерживаемые языки)	Русский	
7.	Минимальные системные требования клиента	Минимальные: Windows 7 ОЗУ – 4гб Процессор - Pentium 4	Требования к ПК, на котором будет использоваться приложение
8.	Технические требования к серверу	В качестве серверного оборудования будет использован Dell PowerEdge R730xd, предназначенный для обработки больших массивов данных Процессор - 1x/2x Xeon E5-2600 v3 “Haswell” Чипсет - Intel C610 Series Chipset ОЗУ – 64гб ОС: RedHat; SuSe Linux	