|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий | |
|  | |
| Кафедра Корпоративных Информационных Систем | |
|  | |
|  | |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРОЕКТНОЙ РАБОТЕ № 1** | |
| **по дисциплине** | |
| **«**Системная и Программная Инженерия**»**  **Тема проекта: «Система контроля жизненных достижений»** | |
|  | |
| Выполнили студенты группы ИКБО-17-18 |  |
| Принял | Гусев К.В. старший преподаватель |

|  |  |  |
| --- | --- | --- |
| Отчет выполнен | «\_\_»\_\_\_\_\_\_\_2020 г. | *(подпись студентов)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2020 г. | *(подпись руководителя)* |

Москва 2020

**Описание проекта**

Проект – “Система контроля жизненных достижений” представляет из себя web-сервис. Сервис представляет собой приложение-агрегатор жизненных достижений, оформленный в виде ориентированного циклического графа, сервис направлен на геймификацию реальных задач. Проект в завершенном виде должен представлять дерево возможных направлений развития личности. Проект должен стать источником вдохновения и мотивации для достижения успехов в реальной жизни.

Дальнейшее описание не является полностью утвержденным проектом, некоторые функции и детали могут меняться, но тем не менее данное описание позволяет примерно понять, что из себя будет представлять проект.

Предположительно корень дерева должен разветвляться на четыре сферы совершенствования личности: физическая, интеллектуальная, духовная, социальная (возможные замены/дополнения: финансовая, культурная). Достижения изначально присутствуют в общей карте достижений в неактивном состоянии, при выполнении пользователем становятся активными. Должна быть продумана блокировка взаимоисключающих достижений. Достижения имеют градацию по сложности выполнения (редкости выполнения). Сервис может помогать в достижении личных ориентиров, даже отсутствующих в изначальном дереве, у пользователя должна быть возможность добавления важных для него достижений в свою личную карту. У пользователя должна быть возможность предлагать достижения для добавления в общую карту, передавая его на модерацию в виде идеи или полностью оформленного достижения. Некоторые достижения могут быть составными, то есть иметь вложенные подзадачи, оставаясь единым узлом графа. Достижение становится активным при выполнении всех/необходимых вложенных подзадач. Сложной задачей является валидация достижений пользователя. Сама идея личной эффективности может существовать при полной честности с самим собой. Но все же могут быть пользователи, которые будут лукавить о достижении целей. Одной из возможных систем контроля является условное подтверждение достижений в виде отчетности (текст/видео/фото). Для полной валидации достижения отчет должен анонимно пройти через других пользователей подтверждающих или опровергающих факт выполнения необходимых для достижения действий, за это проверяющие будут поощряться возможностью вынесения своих достижений на проверку (либо другим способом). Достижения будут оцениваться пользователями по разным критериям (например, увлекательность или полезность), пользователь будет видеть оценку в одном из нескольких вариантов (либо в нескольких вариантах): оценка пользователей, активировавших достижения, оценка всех пользователей. На основании оценки пользователь будет для себя решать какую цель поставить для себя в первую очередь, а от какой воздержаться в связи с нехваткой времени или других ресурсов.

**Роли в проекте и их описание**

Всегов проекте участвуют 5 человек, для каждого подобранна роль, каждому выделены задачи, у каждого свои обязанности, тем не менее каждый из участников может помочь с какой-либо задачей другого участника.

**Состав команды и роли:**

1. Мисаилиди Анастас Аристидович - Team Leader, Project Manager. В обязанности входит: определять весь ход создания и развития продукта; принимать решение, что должно быть сделано, оценивать, соответствует ли реализация концепции продукта или нет; руководить всей командой; отслеживать сроки выполнения задач; оперативно реагировать, если начинаются проблемы с качеством; мотивировать команду работать более эффективно; принимает решения, когда возможны альтернативные варианты при выборе развития продукта в ходе разработки; рисует график прогресса проекта; разработка плана проекта; оценка рисков, контроля функциональных и стоимостных рамок и ежедневная работа с командой на проекте.
2. Цаплин Артемий Денисович – QA(Quality assurance), Technical Writer. В обязанности входит: тестирование программного продукта, не просто внешнюю часть приложения, но и корректность обращения к бэк-энду, быстродействие и отказоустойчивость системы, применяя самые разные методики тестирования; убедиться в том, что приложение работает согласно требованиям и спецификациям; написание плана тестирования; нахождение ошибок в ПО; техническое исследование программы для получения информации о её качестве с точки зрения определённого круга заинтересованных лиц; документированием в рамках решения технических задач, в частности разработки программного обеспечения; написание документа, который бы удовлетворял определённым требованиям.
3. Сергеев Артём Олегович – Tech Leader, Developer. В обязанности входит: заниматься архитектурой; ревью кода; решением технических проблем; координацией работы; написание кода; консультируют других, как писать код; рецензирование кода за другими; принимать решение, как конкретно будет реализована та или иная функциональность в приложении.
4. Шиханов Александр Алексеевич – Database Administrator, Developer. В обязанности входит: выработка требований к базе данных, её проектирование, реализацию, эффективное использование и сопровождение; проектирование базы данных; оптимизация производительности базы данных; обеспечение безопасности в базе данных; обеспечение целостности баз данных; написание кода; консультируют других, как писать код; рецензирование кода за другими; принимать решение, как конкретно будет реализована та или иная функциональность в приложении.
5. Самородов Алексей Александрович – User Interface Designer, User Experience Designer. В обязанности входит: отвечать за функциональность, адаптивность продукта и то, какие эмоции он вызывает у пользователей (чем проще ваш интерфейс, тем проще пользователю получить результат и совершить целевое действие); проектирование интерфейса на основе исследования пользовательского опыта и поведения; работу над графической частью интерфейса (анимацией, иллюстрациями, кнопками, меню, слайдерами, фотографиями и шрифтами); определять цветовую палитру и расположение объектов в интерфейсе (удобно ли попасть по кнопке, правильно ли работает выпадающее меню, удобно ли заполнять форму, хорошо ли читается текст, какое сообщение выдает сайт при том или ином действии).

**Используемые в проекте технологии**

В проекте используются следующие технологии:

* HTML (HyperText Markup Language — «язык гипертекстовой разметки») – стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства. Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам HTTP или HTTPS, в виде простого текста или с использованием шифрования.
* CSS (Cascading Style Sheets — каскадные таблицы стилей) – формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам, например, к SVG или XUL. CSS используется создателями веб-страниц для задания цветов, шрифтов, стилей, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS являлось отделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими шрифт Брайля.
* TypeScript – язык программирования, представленный Microsoft в 2012 году и позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript. TypeScript является обратно совместимым с JavaScript и компилируется в последний. Фактически, после компиляции программу на TypeScript можно выполнять в любом современном браузере или использовать совместно с серверной платформой Node.js. Код экспериментального компилятора, транслирующего TypeScript в JavaScript, распространяется под лицензией Apache. Его разработка ведётся в публичном репозитории через сервис GitHub. TypeScript отличается от JavaScript возможностью явного статического назначения типов, поддержкой использования полноценных классов (как в традиционных объектно-ориентированных языках), а также поддержкой подключения модулей, что призвано повысить скорость разработки, облегчить читаемость, рефакторинг и повторное использование кода, помочь осуществлять поиск ошибок на этапе разработки и компиляции, и, возможно, ускорить выполнение программ. Планируется, что в силу полной обратной совместимости адаптация существующих приложений на новый язык программирования может происходить поэтапно, путём постепенного определения типов. TypeScript возник из-за предполагаемых недостатков JavaScript в крупномасштабных приложениях как в Microsoft, так и у прочих пользователей JavaScript.[10] Проблемы с разработкой сложных программ на JavaScript привели к необходимости облегчения разработки компонентов языка. Синтаксически, TypeScript очень похож на JScript .NET, очередную реализацию Microsoft языкового стандарта ECMA-262, обеспечивающего поддержку статической типизации и классических объектно-ориентированных возможностей языка, таких как классы, наследование, интерфейсы и пространства имен.
* Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций. Python поддерживает структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.
* SQL (structured query language — «язык структурированных запросов») – декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных. Является, прежде всего, информационно-логическим языком, предназначенным для описания, изменения и извлечения данных, хранимых в реляционных базах данных. SQL считается языком программирования, в общем случае (без ряда современных расширений) не является тьюринг-полным, но вместе с тем стандарт языка спецификацией SQL/PSM предусматривает возможность его процедурных расширений.
* GIT – распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года. На сегодняшний день его поддерживает Джунио Хамано. Программа является свободной и выпущена под лицензией GNU GPL версии 2. По умолчанию используется TCP порт 9418. GitHub мы используем для слаженной работы и отслеживания изменений. Ссылка на репозиторий проекта: [https://github.com/kazzart/Achievy](https://vk.com/away.php?to=https%3A%2F%2Fgithub.com%2Fkazzart%2FAchievy&cc_key=)

**Ход работы**

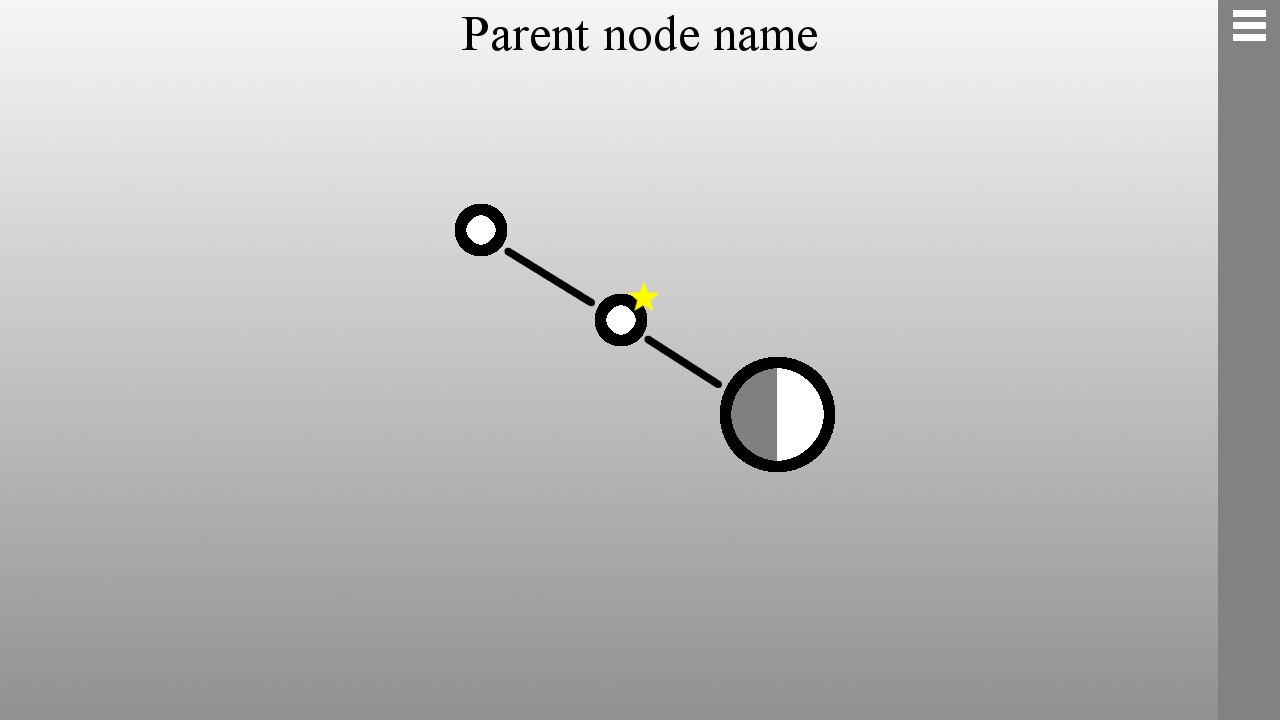
Был подобран данный дизайн, но в дальнейшем дизайн может измениться или преобразоваться: 

Рис. 1 – Дизайн основного экрана с деревом достижений

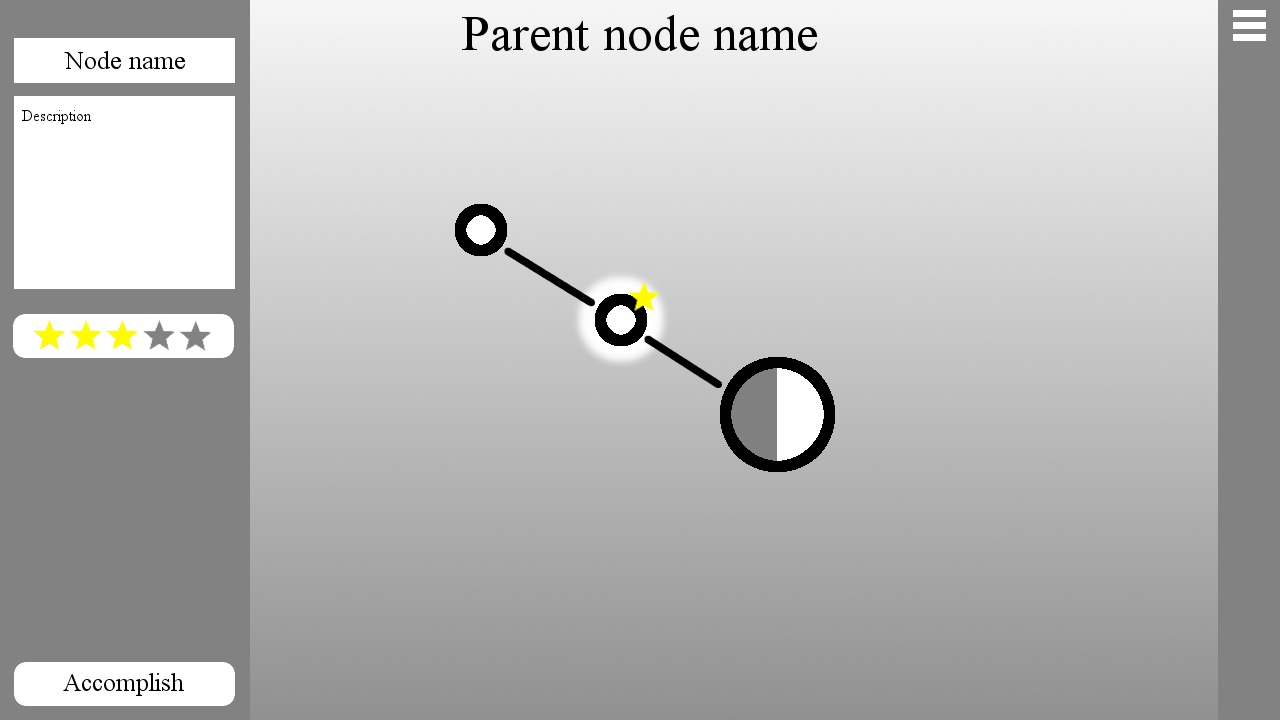


Рис. 2 – Дизайн основного экрана с деревом достижений (выбран один элемент дерева)

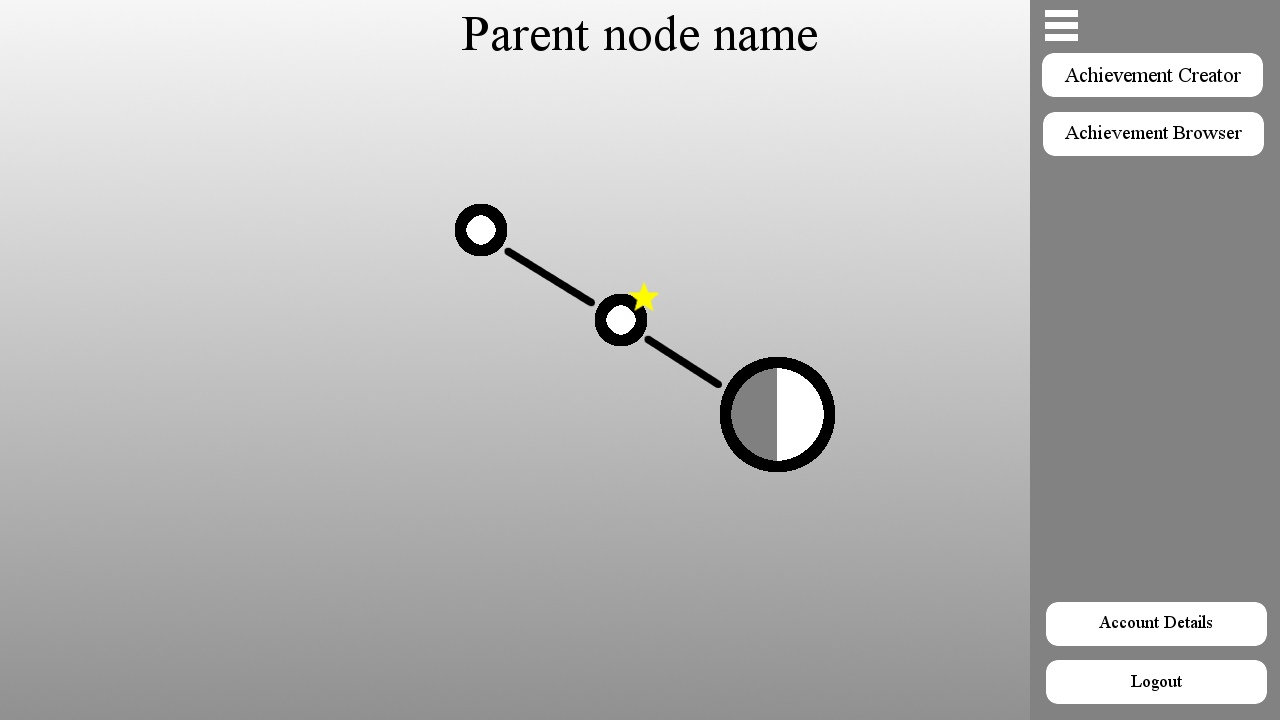


Рис. 3 – Дизайн основного экрана с деревом достижений (открыто меню)

Так же было начато изучение требующихся нам технологи и выбран дальнейший функционал проекта.

**Дальнейшие действия в реализации проекте**

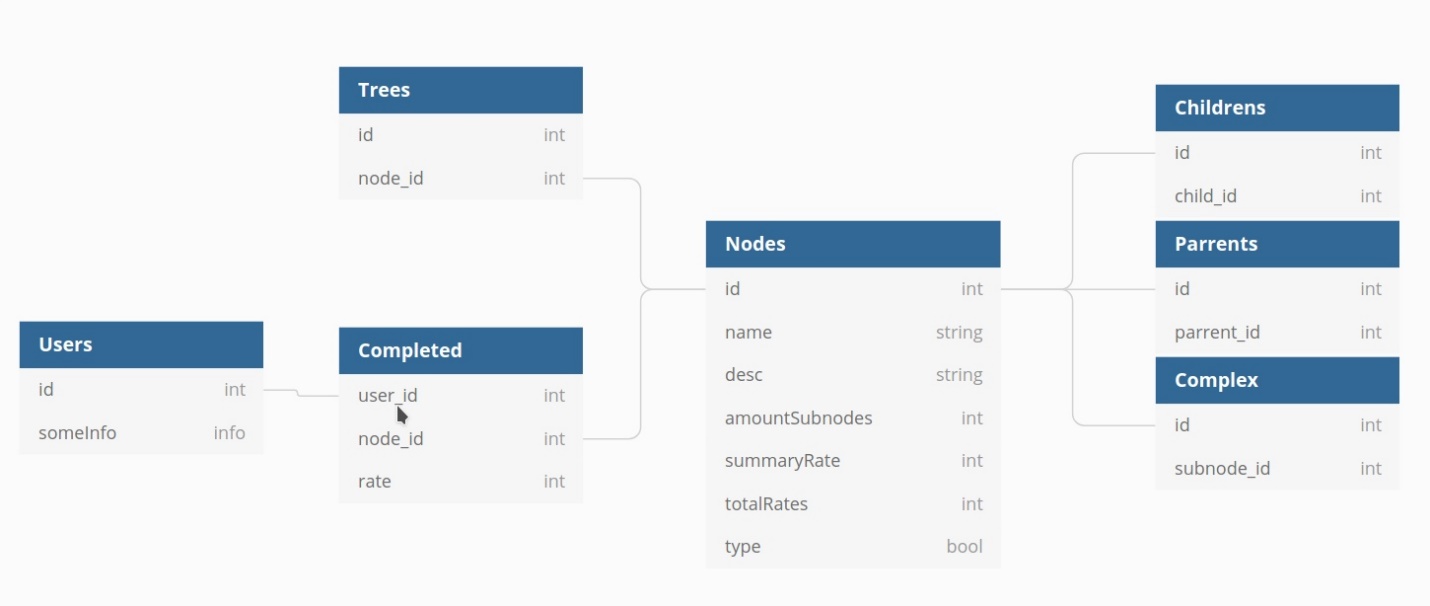
В ближайшей перспективе будет реализованная данная структура: 

Рис. 4 – База данных проекта

Также будут реализованы ключевые возможности web-сервиса.