



Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Розрахункова графічна робота

З дисципліни «Інтеграційні програмні системи»



Виконали:

Студенти IV-го курсу
Групи ІО-42 ФІОТ
Байдиков Микола
Василенко Сергій
Вернер Анна

Перевірив:

Мазур Роман Федорович

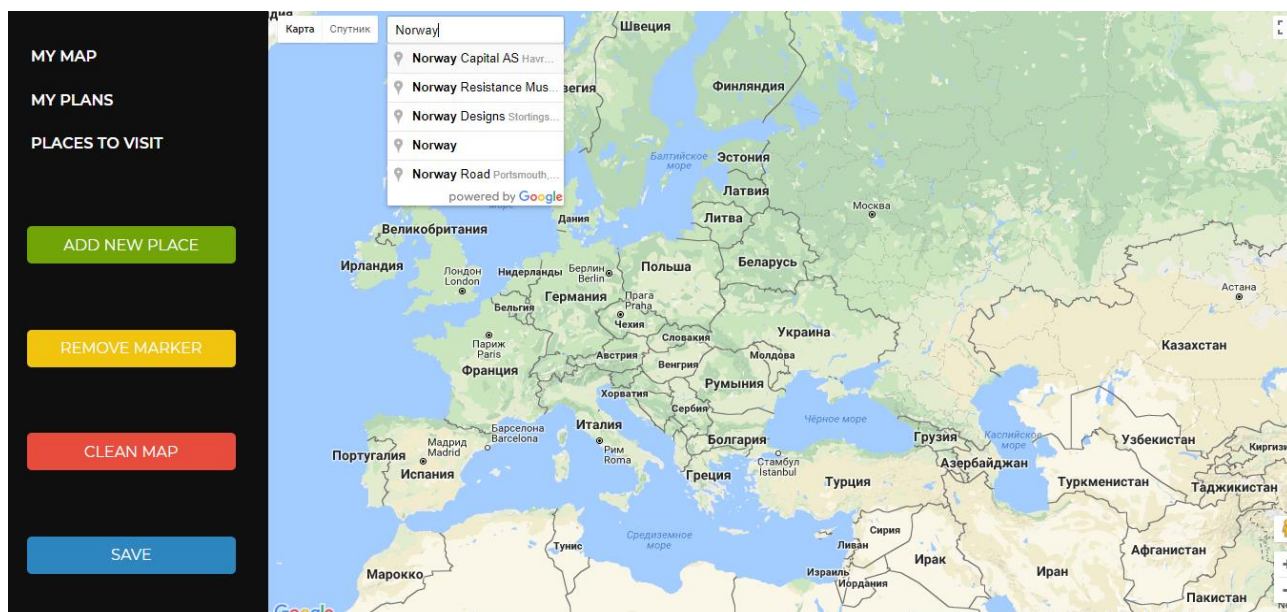
Київ 2017

1. Краткое описание проекта

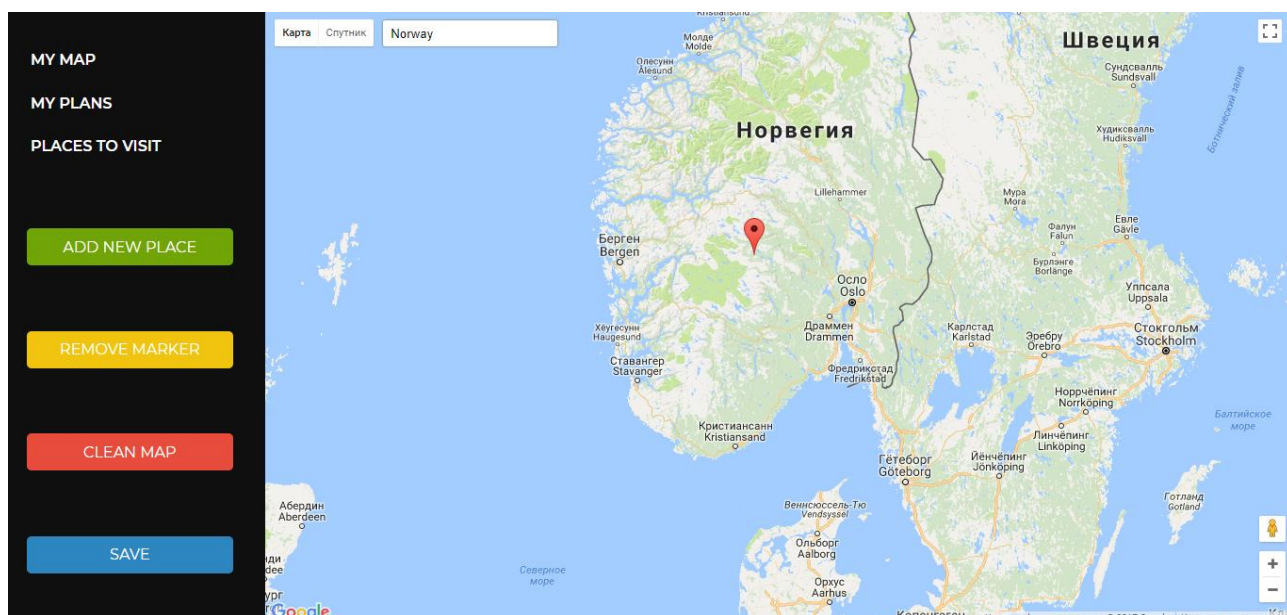
Система предназначена для пользователей, которые хотят путешествовать или интересуются таким видом досуга. Кроме того, система обеспечивает возможность планирования будущих путешествий и отслеживания истории посещенных мест, введенных в систему пользователем из списка, предоставленного средствами Google Maps API.

В данном сервисе представлены такие функции работы с картой как:

1. Выполнение поиска при помощи поля с подсказками.

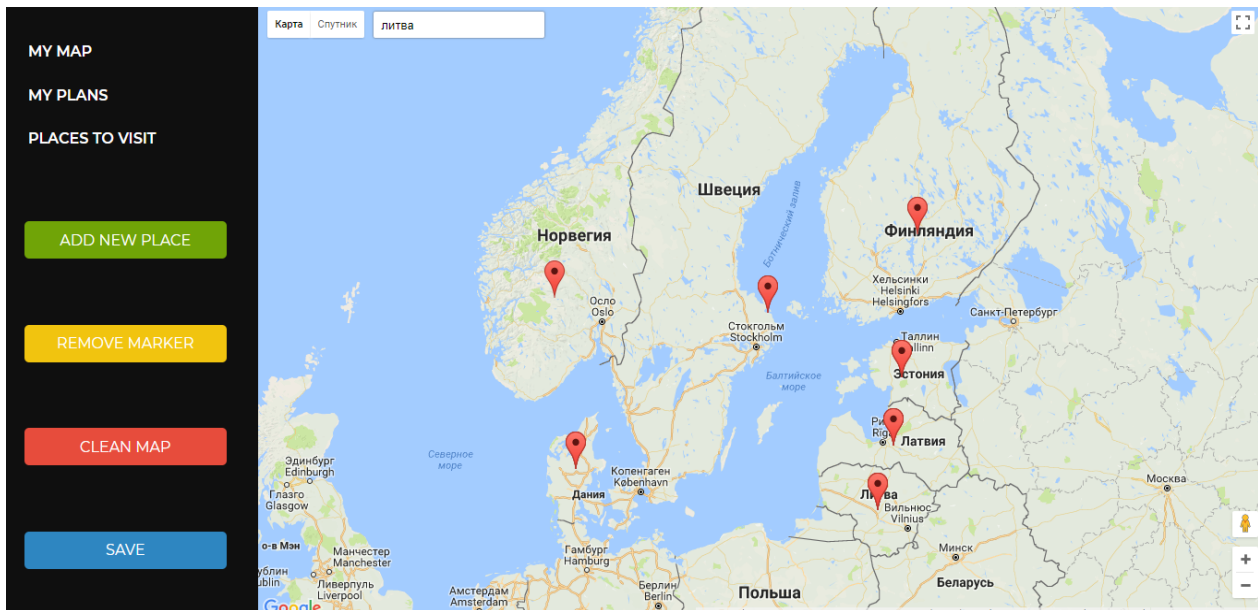


2. Возможность добавить маркер места на карту.

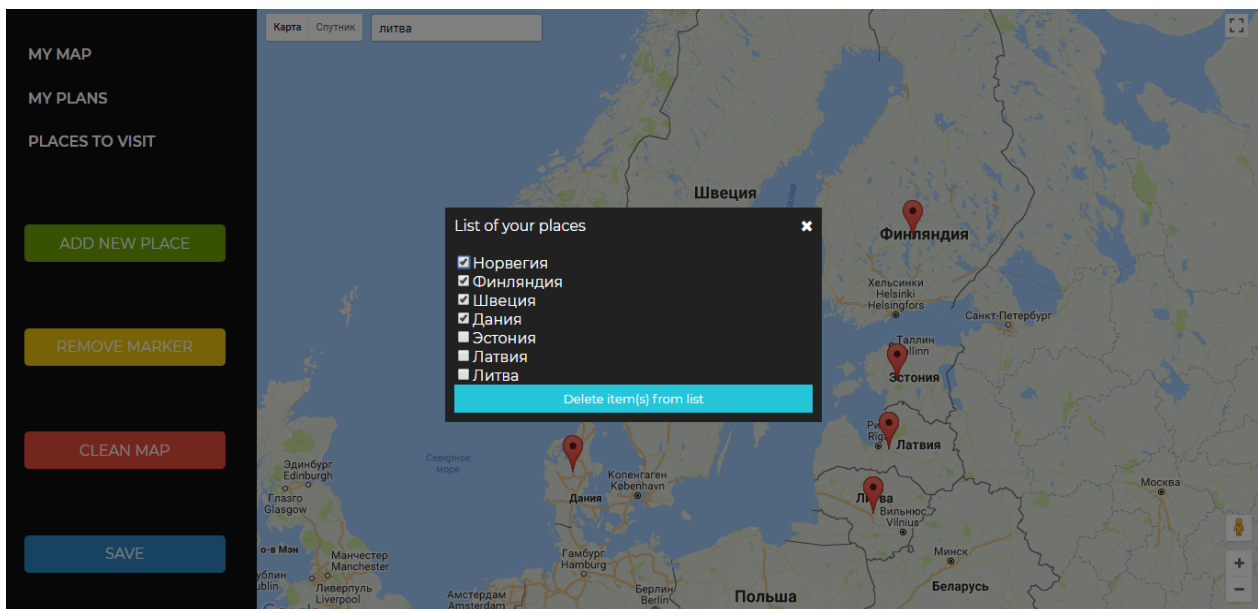


3. Удаление маркера или группы маркеров с карты и просмотр списка уже добавленных мест.

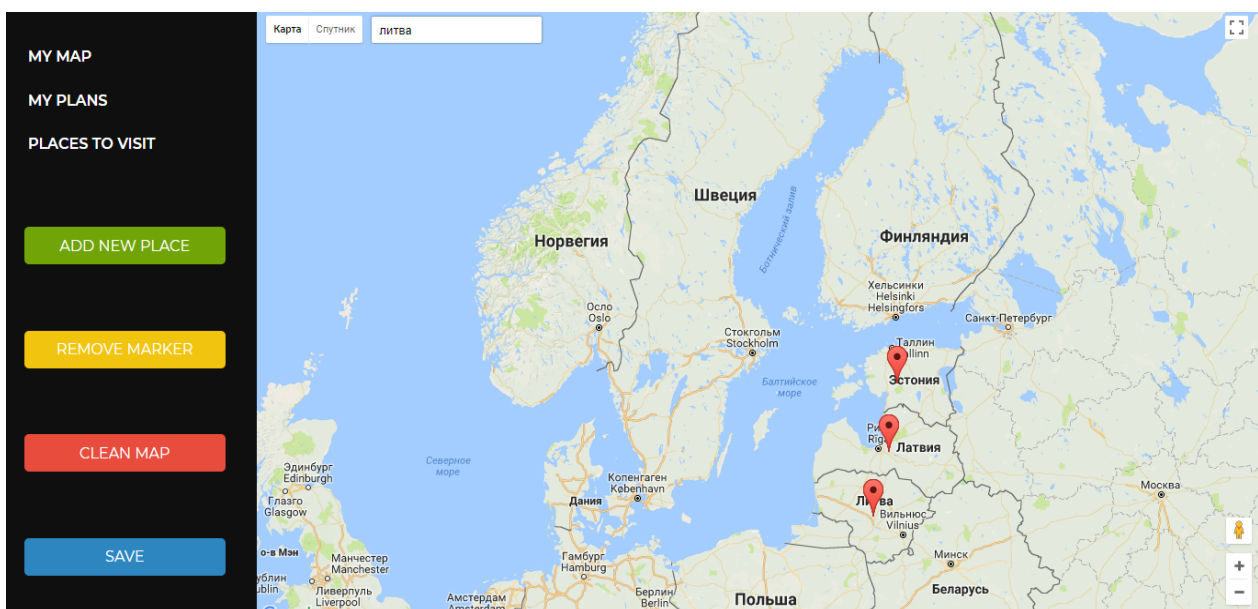
Входные данные:



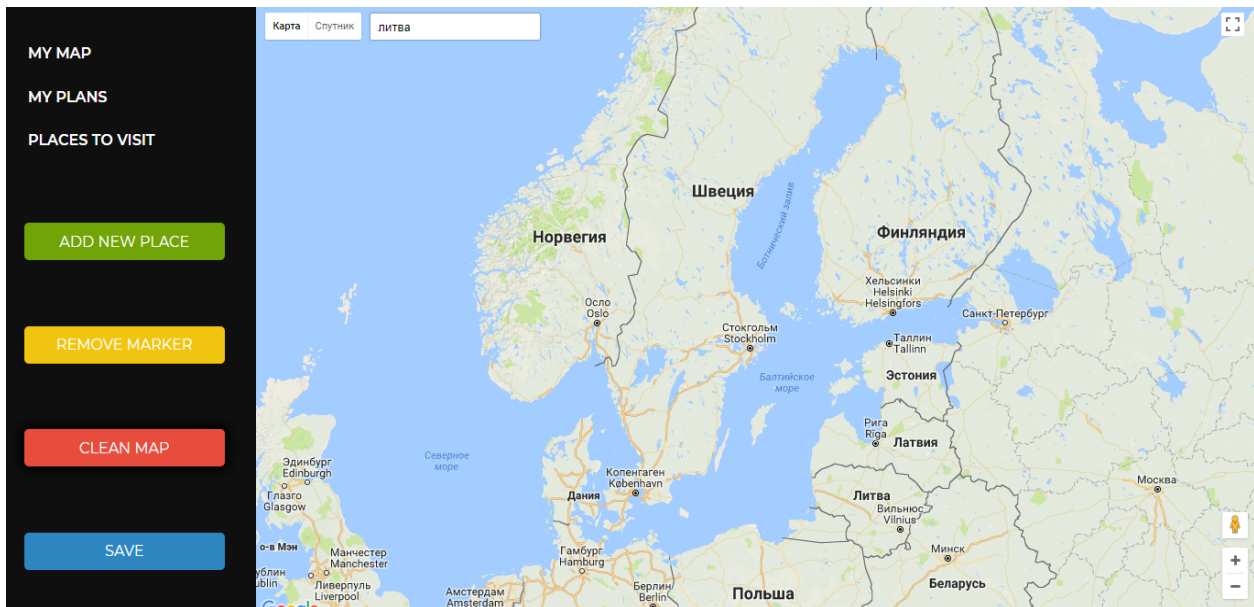
Немного магии по выбору мест для удаления:



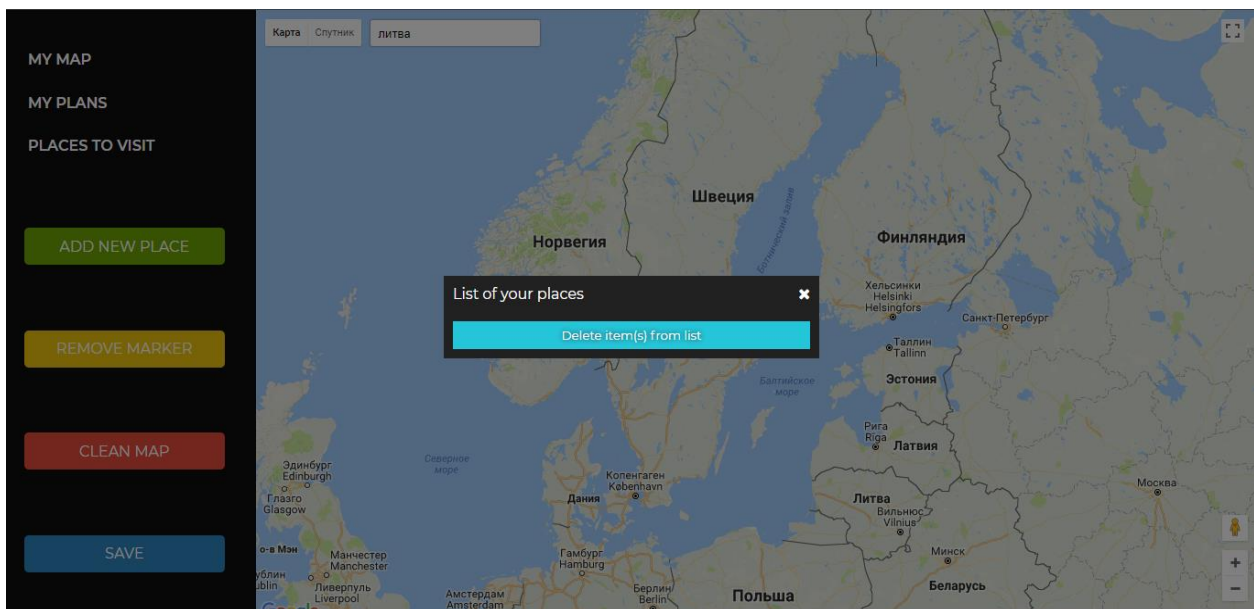
Выходные данные:



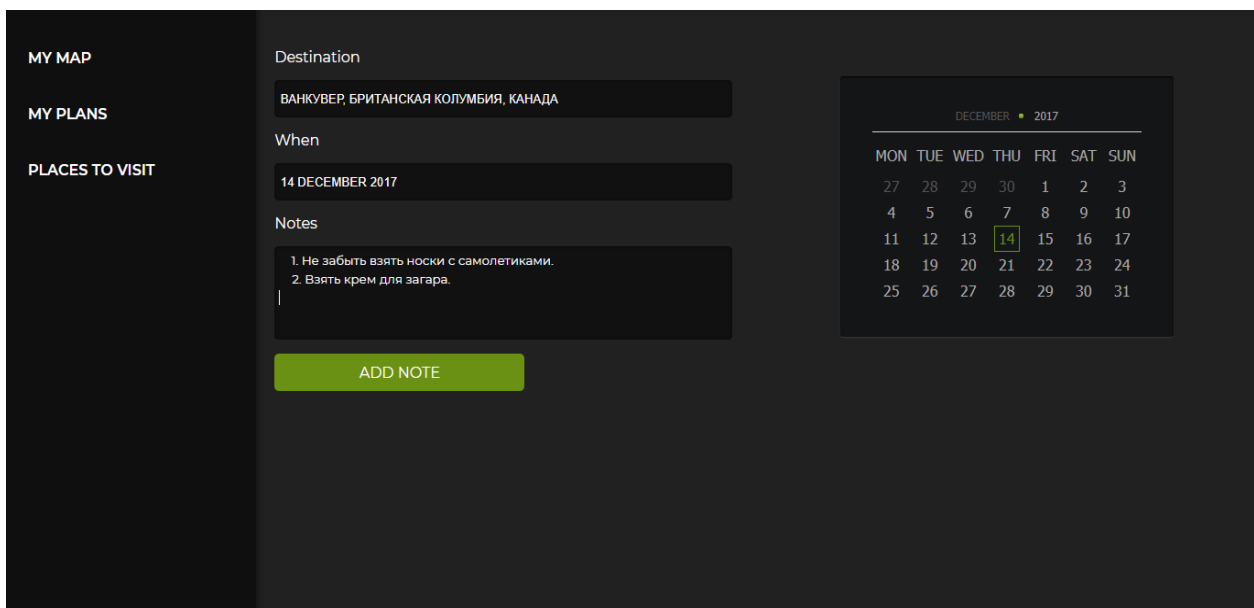
3. Очистить карту.



Одновременно с этим очищается список мест



4. Возможность добавить заметки



2. Системы сборки

Так как проект содержит в себе несколько интегрирующийся частей, а именно разработки под платформу iOS и веб (back-end и front-end) необходимо было использовать три различные системы сборки. В качестве систем сборки были выбраны следующие:

IOS	Xcode
Back-end	Maven
Front-end	Yarn

Maven – это один из трёх (Ant, Gradle) основных инструментов, которые используются для выполнения сборки Java проекта, а именно непосредственной компиляции, последующего создания jar-файла, дистрибутива программы, а также генерации документации. Выбор сделан в пользу Maven, основываясь на ряде его преимуществ, а именно: сборка проекта платформонезависимая и выходной файл будет одинаковым для любой из них; удобное управление зависимостями позволяет разрешить конфликты версий подключаемых ресурсов и затем при необходимости легко переходить к новым обновлениям подключаемых ресурсов, к примеру, библиотек; сборка проекта возможна из командной строки, что часто бывает необходимым для осуществления автоматической сборки в процессе непрерывной интеграции. Одно из основных отличий Maven в том, что цели жизненного цикла сборки уже определены, вы только можете вызывать определённую цели или же настраивать её под свои нужды. Maven конфигурируется файлом pom.xml. Основные среды разработки на java легко открывают проекты, которые собираются при помощи Maven. При этом зачастую проект настраивать не нужно - он сразу готов к дальнейшей разработке. Как следствие - если с проектом работают в разных средах разработки, то Maven удобный способ хранения настроек. Настраиваемый файл среды разработки и для сборки один и тот же - меньше дублирования данных и соответственно ошибок. Помимо вышеперечисленного, в текущем проекте система сборки используется для превращения jar-файла в образ docker, а также подразумевает запуск unit и integration тестов.

Yarn – этот менеджер пакетов используется в качестве системы сборки для JavaScript. Выбор сделан в пользу Yarn, который непосредственно разработан на базе менеджера пакетов npm, поскольку помимо того, что он работает быстрее благодаря параллельному выполнению задач во время установки пакетов в отличии от вышеупомянутого менеджера пакетов. Все зависимости отслеживаются в package.json, с помощью которого Yarn и npm отслеживают зависимости, номера версий не указываются точно, вместо этого задается их диапазон. Таким образом, появляется возможность задать определенную основную или вспомогательную версию, при этом позволяя npm установить последний патч с исправлениями каких-либо багов. Стратегия, выбранная npm может привести к тому, что на двух машинах с идентичными файлами package.json будут установлены различные версии пакетов, что может привести к появлению багов. Чтобы избежать несоответствия версий, установленная версия закрепляется в файле блокировки. При каждом добавлении модуля создается файл yarn.lock. Таким образом возможным является гарантирование того, что на другой машине будет установлен тот же самый пакет; при этом у нас сохраняется диапазон

допустимых версий в файле package.json. Команда yarn install только устанавливает зависимости, перечисленные в файле yarn.lock или package.json. В разрабатываемом проекте при помощи Yarn происходит запуск unit тестов, а также запуск сервера, на котором находится фронтенд.

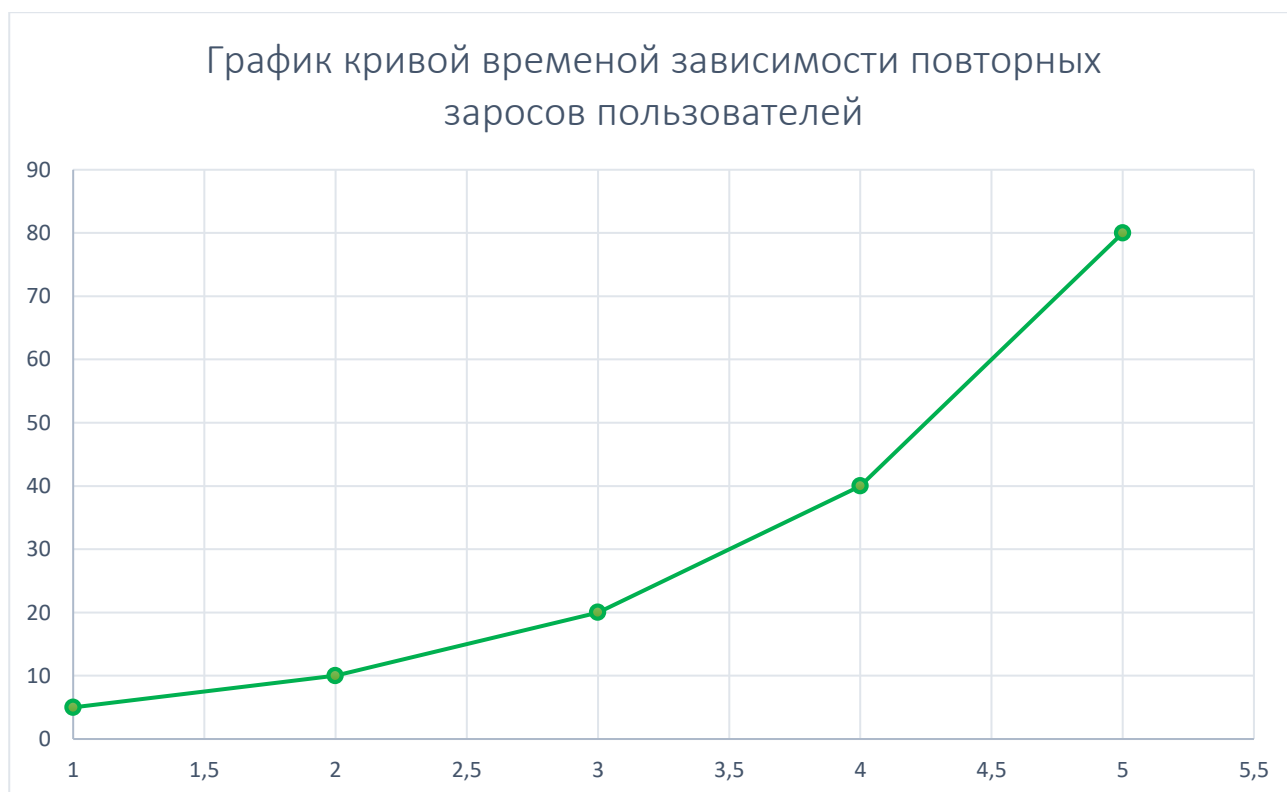
Xcode – среда разработки от Apple. На данный момент является единственным полностью независимым инструментом для написания, сборки и публикации iOS приложений. Все конфигурации вносятся в файл проекта, для которого в среде есть удобный графический интерфейс. Все зависимости управляются с помощью библиотеки cocoapods и хранятся в Podfile. С помощью Xcode можно собирать и запускать сам проект, а также unit и интеграционные тесты. Также предоставляет команды для сборки проекта из командной строки, благодаря чему его можно использовать в системах непрерывной интеграции.

3. Описание задач, которые выполняются в процессе непрерывной интеграции на сервере:

1. Запуск docker image с базой данных mariadb.
2. Проверка checkstyle back-end части.
3. Build image back-end части.
4. Запуск docker image back-end части.
5. Проверка checkstyle front-end части.
6. Запуск js тестов.
7. Запуск интеграционных тестов.

4. Экспоненциальная выборка

В процессе экспоненциальной выборки получение следующего времени повторения запроса производится посредством умножения предыдущего полученного значения на два.



Nick Baidikov



IOS
part developer

Group IO-42
YO-ANTIHYPE-TEAM

Serhii Vasilenko



Back-end
part developer

Group IO-42
YO-ANTIHYPE-TEAM

Anna Verner



Front-end
part developer

Group IO-42
YO-ANTIHYPE-TEAM