

УДК 001.895+336.581

РАЗРАБОТКА ОНЛАЙН ПЛАТФОРМЫ ОЦЕНКИ И ФИНАНСИРОВАНИЯ ИННОВАЦИОННЫХ ПРОЕКТОВ



В.Ф. Алексеев

Доцент кафедры проектирования информационно-компьютерных систем БГУИР, кандидат технических наук, доцент



Д.В. Лихачевский

Декан факультета компьютерного проектирования БГУИР, кандидат технических наук



Г.А. Пискун

Доцент кафедры проектирования информационно-компьютерных систем БГУИР, кандидат технических наук, доцент

*Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь
E-mail: alexvikt.minsk@gmail.com, piskunbsuir@gmail.com, likhachevskiyd@bsuir.by*

Аннотация. На базе современных методов показан подход к проектированию онлайн платформы для оценки финансирования инновационных проектов.

Ключевые слова: инновации, инновационный проект, B2B, UML, инвестор, инноватор, информационная модель.

Введение. В современной экономике роль инноваций значительно возросла. Без применения инноваций практически невозможно создать конкурентоспособную продукцию, имеющую высокую степень наукоемкости и новизны. Таким образом, в рыночной экономике инновации представляют собой эффективное средство конкурентной борьбы, так как ведут к созданию новых потребностей, к снижению себестоимости продукции, к притоку инвестиций, к повышению имиджа производителя новых продуктов, к открытию и захвату новых рынков, в том числе и внешних [1-5].

Одной из самых динамичных, гибких и креативных структур экономики является предпринимательство, именно данный факт делает его идеальной средой для развития инновационных идей и проектов. Именно предпринимателей отличает инновационный тип мышления.

Динамика развития инновационных предприятий имеет значительную зависимость от экономической политики государства и созданной им инновационной инфраструктуры. Не последнюю роль играет и доступ к инвестициям, так как большинство инноваций требуют на начальной стадии развития значительного финансирования, средствами для осуществления которого, как правило, не обладают малые и средние предприятия.

Разработка структуры и функций веб сервиса. Приложение будет нести B2B характер. Главной задачей сервиса будет, с одной стороны, получение инвестиций инноваторами, с другой стороны, выгодное вложение в инновационные проекты инвесторами, либо же поддержка интересных проектов обычными пользователями в обмен на будущий результат проекта.

Чтобы привлечь внимание инвесторов и генераторов инновационных проектов необходимо разработать специализированный и удобный в использовании сервис, который в перспективе может стать основным инструментом проведения политики по финансированию инновационных проектов.

Проектирование системы начинается с постановки задач и определения основных её возможностей. Проектирование осуществляется с помощью UML (Unified Modeling Language) диаграмм.

Сервис разрабатывается для следующих трех основных группы пользователей: инноватор, инвестор и последователь. К последователям будем относить обычных пользователей, не подтвердивших статус инвестора, и не владеющие собственными проектами.

На основе анализа существующих методов финансирования и стимулирования инновационных проектов, выделим основные задачи, которые необходимо решить для достижения поставленной цели:

- с точки зрения инноватора сервис должен предоставлять возможность максимально привлекательно предоставить суть своей идеи и обосновать ее экономическую эффективность и востребованность;

- с точки зрения инвестора и последователя сервис должен предоставлять возможность многофункционального поиска интересующих его проектов. Должен осуществляться поиск самых рейтинговых проектов, поиск по имени, поиск по категориям;

- с точки зрения последователя, должна быть возможность обсудить и поддержать проект материально, подразумевается, что за материальную помощь инноватор может отблагодарить последователя;

- с точки зрения инвестора, должна быть возможность связаться с инноватором, для более детального обсуждения объема и условий предоставления инвестиций;

- реализовать возможность создания и редактирования профилей пользователей;

- должна быть реализована функция информирования клиентов сервиса через электронную почту.

Кроме того, программный продукт должен отвечать ряду технологических требований:

- приложение должно иметь web-ориентированную архитектуру, с толстым клиентом, поскольку оно предназначено для широкого использования, и должно работать быстро, иметь привлекательный внешний вид;

- при реализации проекта должны использоваться только современные технологии.

Следуя последнему требованию, необходимо определить набор технологий, позволяющий наиболее быстро и качественно реализовать поставленные задачи.

Выбор и обоснование технологии реализации. Данный проект может быть реализован в рамках слоистой серверной архитектуры поскольку это дает возможность любой функционал вписать в одни рамки. Такая архитектура также дает возможность запросто изменять функционал в одном слое, ничего не меняя в другом. Таким образом, можно выделить бизнес слой, DAO слой и слой моделей данных. Для реализации *UI* части проекта можно использовать *MVVC* архитектуру на стороне клиента, тем самым выбор был сделан в сторону толстого клиента, что позволит значительно улучшить скорость работы, и даст преимущества в широком выборе *UI* компонентов. Для организации общения между клиентом и сервером можно выбрать *REST* сервисы, которые являются визуально более понятными для пользователя и четко завязаны на предметную область приложения, так как имеют прозрачные правила написания.

Для работы с базой данных можно использовать, например, *Hibernate*, что должно значительно облегчить *CRED* операции. Несмотря на то, что *Hibernate* имеет много плюсов, он все-таки является более медленным, чем *native sql*, поэтому, необходимо учитывать возможность увеличения базы данных при ее проектировании. Для реализации быстрого поиска было решено использовать индексы, а именно технологию *Elasticsearch*.

Для управления зависимостями объектов в приложении можно использовать *Spring IoC*. Для простоты разработки *REST* сервисов используем фреймворк *Jersey*. Для реализации *front-end* части приложения используем *AngularJs*. Это также должно придать приложению более привлекательный и современный вид, так как *AngularJs* обладает массой возможностями в об-

ласти построения внешнего вида приложения. Фреймворк работает с *HTML*, содержащим дополнительные пользовательские атрибуты, которые описываются директивами, и связывает ввод или вывод области страницы с моделью, представляющей собой обычные переменные *JavaScript*. Значения этих переменных извлекаются из динамических *JSON*-данных. Главным преимуществом данного фреймворка является двусторонняя привязка данных для динамического контента, что позволяет автоматически синхронизировать модель и представление. В результате *AngularJS* уменьшает роль *DOM*-манипуляций и улучшает тестируемость.

Рассматривая задачу информативности страницы инновационного проекта, которая в тоже время должна быть понятной и неперегруженной. Авторами предложена идея разделить все информации на табы, и предоставить максимальное количество подсказок.

Для мультимедийного контента было решено не хранить видео и картинки на сервере сервиса. Для хостинга видео выбран сервис *Youtube*, для картинок использовался *DUDE*. Время жизни проекта будет контролироваться каждую ночь. Для этого предполагается использовать *Spring Quartz*.

Для отображения местоположения бизнеса и клиента можно использовать *Google Maps API*. Надо заметить, что *Google Maps* является платным *API*, если число запросов в месяц превышает 10000.

Для реализации *UI* части приложения используется фреймворк *AngularJs*, который позволяет легко внедрять одностраничные приложения, что позволит легко реализовать много шаговые интерфейсы создания проекта, юзера или процесса дотации проекта. Необходимо отметить, что все общения с сервером происходит по технологии *AJAX*, что позволяет организовывать общение клиента и сервера без перезагрузки страницы целиком. Поэтому практически все запросы, в разработанном сервисе работают с применением технологии *AJAX*, что позволило снизить в значительной мере планируемую нагрузку на сервер и увеличить скорость работы системы.

Учитывая, что разработанная система должна обрабатывать значительный объем данных, очевидно, что значительные ресурсы будут тратиться на операции с базой данных. *Hibernate* обладает неплохими возможностями для увеличения производительности при работе с большими объемами информации. Он предлагает такие решения как «ленивая инициализация», кеши двух уровней и другие.

Также для удобства сборки приложения и простоты работы с библиотеками можно использовать *Maven*, *Grunt* – для минимизации файлов и реализации различных утильных задач, для уменьшения зависимостей от сторонних *js* библиотек было решено использовать *Bower*.

Таким образом, были рассмотрены способы решения основных задач, поставленных в предыдущей главе.

Построение информационной модели. Рассмотрим информационную модель системы. Она визуализирована на рисунке 1, с помощью средств приложения *WorkBench*. В информационной модели были выделены сущности, которые тесно переплетаются с понятиями, выделенными при анализе предметной области приложения. Двумя ключевыми сущностями являются *user* и *project*. У пользователя может быть несколько проектов, поэтому данные сущности связаны связью один ко многим.

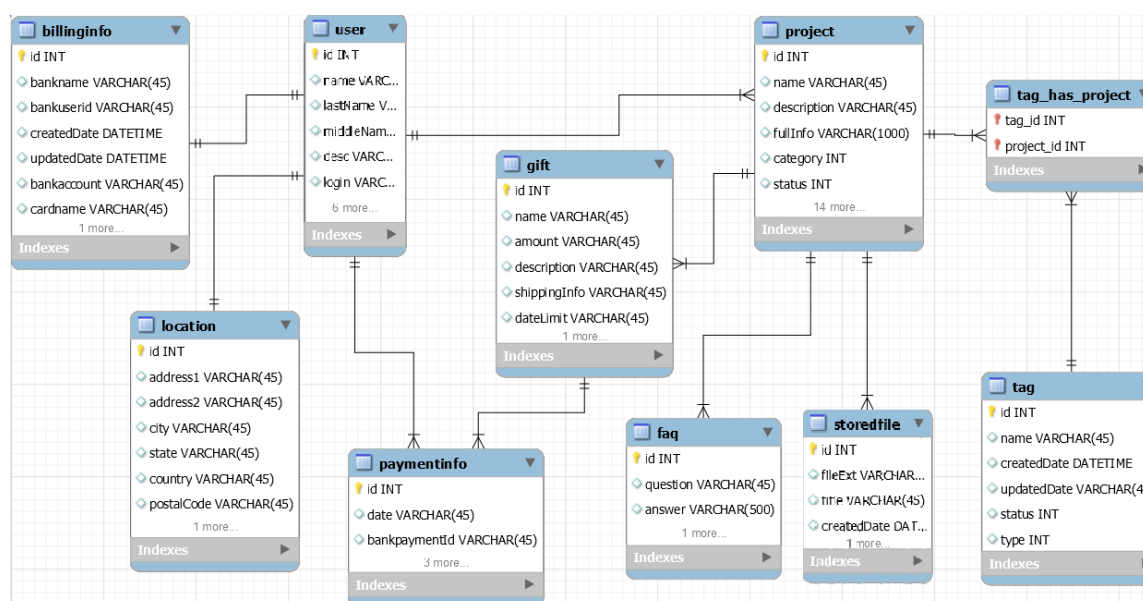


Рисунок 1. Физическая информационная модель системы

Пользователь связан с таблицами *billinginfo* и *location* связью один к одному. В таблице *billinginfo* хранится информация о счете, на который пользователь будет получать средства на свой проект. В *location* хранится информация о местоположении пользователя. Таблица *project* имеет связь один ко многим с таблицей *storedfile*, в которой будет храниться информация о всех связанных с проектом фото и видео файлов и их тип, также здесь будут помещаться записи о документах, загруженных к данному проекту. Важным и универсальным инструментом хранения информации, связанной с проектом, является таблица *tag*, которая связана с таблицей *project* связью многие со многими. Такая связь была выбрана поскольку предполагается, что один и тот же *tag* можно будет повесить на множество проектов, и у проекта может быть множество тэгов. Данный механизм универсально позволяет хранить информацию различного типа, конкретно не связанную с проектом, но имеющую к нему отношения. Также таблица *project* связана с таблицей *faq* связью один ко многим, в которой хранится информация о часто задаваемых вопросах и ответах на них.

В сущности *gift* хранятся данные о поощрениях для дарителей. У каждого проекта может быть много такого рода поощрений, именно поэтому таблица проектов и поощрения связаны связью один ко многим. При дарении средств, сведения об этой операции записываются в таблицу *paymentinfo*, которая связана с таблицами *user* и *gift*.

Таким образом, информационная модель полноценно отображает предметную область системы. Надо заметить, что при проектировании БД было обращено особое внимание на производительность ее работы с большим количеством данных, что в современных системах имеет особую актуальность.

Разработка и построение UML диаграмм. Для проектирования вариантов использования приложения и последующей их реализации воспользуемся диаграммой вариантов использования. Выделим три типа пользователей системы: инвестор, инноватор и администратор. Рассмотрим основные функции, которые должны быть предоставлены выделенным пользователям, начиная с инноватора.

Инноватор, как и все другие пользователи, должен иметь возможность регистрироваться в системе. Предполагается, что пользователь сможет зарегистрироваться в сервисе, используя сторонние системы, такие как *Facebook*, *Vkontakte* или *Google*. Данная возможность облегчит процесс регистрации и доступа к функционалу сервиса, что должно положительно сказаться

на удобстве пользования сервисом. Основная же возможность, которая предоставляется инноватору – это возможность привлекать внимание и средства к своему проекту через создание проекта. Кроме подробнейшего описания проекта, можно будет разместить видео и фото, прикрепить документы. Также инноватор может предоставить мелким инвесторам поощрения за определенный объем инвестиций. К примеру, за минимальный объем может предоставляться некая сувенирная продукция с элементами логотипа проекта, при более крупном – можно предоставлять сам продукт инновация в качестве поощрения. При крупном же инвестировании, онлайн сервисом не обойтись, так как крупные инвестиции не возможно провести как дарение, и для этого должны заключаться договоры, формы входа инвестора и другие вопросы. Инноватор должен иметь полноценные функции управления своими проектами.

Любой посетитель сервиса является потенциальным инвестором. Поэтому все пользователям должно предоставляться возможность искать проекты. Поиск должен осуществляться как по фразе так и по категориям проектов. Любой пользователь может поддержать проект, выбрав любой вид поощрения и переведя средства инноватору. Так же должны быть возможность оставить комментарии к проекту. Для крупных инвесторов или государственных должны быть возможность связаться с инноватором или оставить заявку для обратной связи.

Для такого вида сервиса, который предполагается реализовать, необходим администратор, который смог бы модерировать проекты, редактировать права доступа и выделять проекты в результатах поиска, чтобы они становились более заметными.

При разработке web-ориентированного сервиса бизнес логика сервера была реализована на основе слоевой архитектуры, а клиент решено было реализовать с помощью *AngularJs*. Для наглядности реализуемой архитектуры была использована диаграмма последовательности, представленная на рисунке 2, которая иллюстрирует общую последовательность обработки запросов к web-серверу на основе обработки поискового запроса.

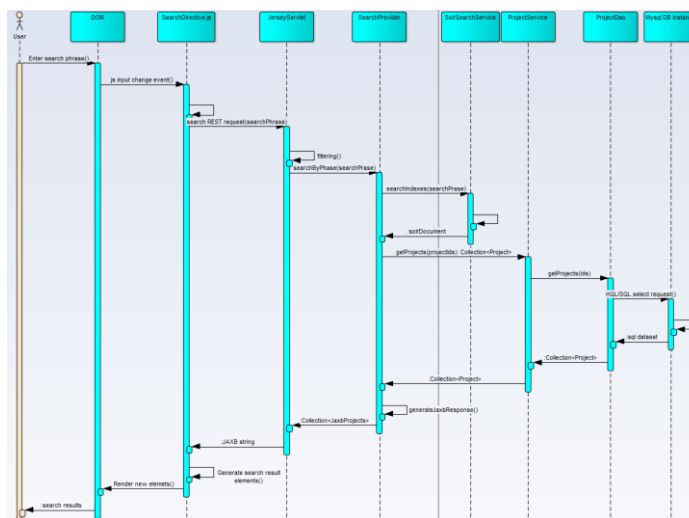


Рисунок 2. Диаграмма последовательности обработки поискового запроса

Рассмотрим более подробно последовательность происходящих событий. Пользователь заполняет форму поиска. Фактически он изменяет состояние *DOM* элемента, в результате чего генерируется *change event* поисковой формы ввода. Данное событие слушается директивной *AngularJs*, которая формирует поисковый *get* запрос и отправляет его *REST* сервису. На сервере запрос передается *JerseyServlet*. *JerseyServlet* осуществляет *mapping* к конкретному *rest service*, в нашем случае управление передается *SearchProvider*, который ответственный за обработку поисковых запросов. Вначале он вызывает *SolrSearchService*, который производит поиск по введенной фразе проектов в индексах. После чего возвращает отсортированные по

релевантности *id* проекты. Далее *SearchProvider* по данным *id* достает полные объекты проектов, из которых генерируется *jaxb response*, который возвращается сервером клиенту. На данный *ajax request* браузер в директиве *SearchDirective* был повешен *callback*, который и вызывается при возврате *response*. После чего директива генерирует панели результата поиска, которые вставляются в *DOM* дерево и отрисовываются браузером.

Сложная структура классов вызывает необходимость в разработке диаграммы компонентов, представленной в приложении Е. Для того чтобы не перегружать диаграмму, но дать представление о структуре разрабатываемого приложения, было решено отобразить только основные классы приложения, отвечающие за работу с моделью Проект, не углубляясь в мелочи.

Для упрощения использования приложения все необходимые компоненты web-приложения упаковываются в war-архив, что позволяет легко запускать приложение на разных машинах, на которых доступен web контейнер. Диаграмма развертывания, представленная в приложении Ж как раз и отражает запуск приложения при условии наличия необходимого ПО (так, к примеру, для сервера необходимо наличие СУБД *MySql*, *JDK 8* и *Tomcat 8*). Также отметим, что так как приложение имеет толстый клиент, реализованный на *AngularJs*, то поддержка *javascript* должны быть включена в браузере клиентской машины.

Также на начальном этапе разработки, необходимо четко сформировать представление о жизненном цикле основных объектов, задействованных в системе, в нашем случае – объекта Проект. Для этого была построена диаграмма состояний, представленная на рисунке 3.

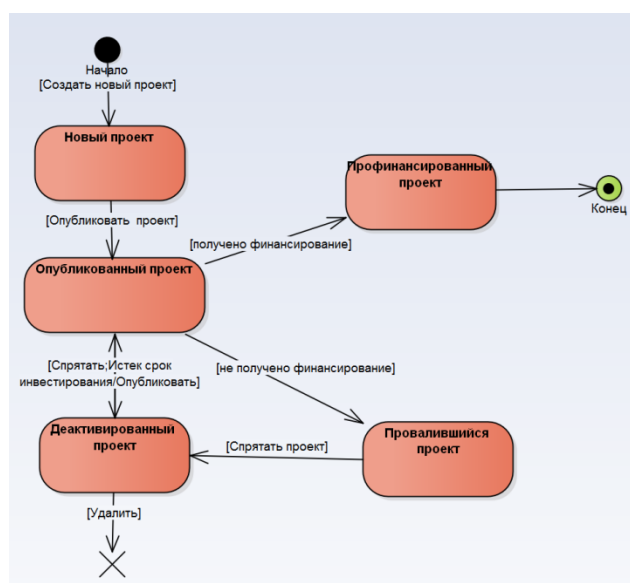


Рисунок 3. Диаграмма состояний объекта Проект

После первого сохранения мы имеем новый объект проекта, после чего он может многократно редактироваться, однако чтобы стать доступным для поиска он должен быть опубликован. Объект может быть скрыт (деактивирован) либо пользователем, либо автоматически при истечении срока действия. Если проект профинансирован удачно, то он меняет статус на профинансированный, если же наоборот – то на провалившейся, после чего он может быть скрытым. Скрытый объект может быть удален.

Таким образом, мы рассмотрели основные моменты, касающиеся планирования и проектирования, разрабатываемого приложения, что позволит сократить время разработки, благодаря сокращению времени на архитектурные вопросы при кодировании.

Список литературы

- [1]. Алексеев, В. Ф. Автоматизация учетных и управленческих решений деятельности организации на базе платформы 1С:Предприятие. / В. Ф. Алексеев, Г. А. Пискун, В. А. Мишута, Е. С. Харитончик // Международная научно-техническая конференция, приуроченная к 50-летию МРТИ-БГУИР (Минск, 18-19 марта 2014 года) : материалы конф. В 2 ч. Ч. 2. – Минск, 2014. – С. 219-220.
- [2]. Алексеев, В. Ф. Особенности построения системы управления взаимоотношениями с клиентами в деятельности банка. / В. Ф. Алексеев, Г. А. Пискун, В. А. Мишута, Е. С. Харитончик // Международная научно-техническая конференция, приуроченная к 50-летию МРТИ-БГУИР (Минск, 18-19 марта 2014 года) : материалы конф. В 2 ч. Ч. 2. – Минск, 2014. – С. 217-218.
- [3]. Инновации и экономический рост / Христенко В.Б. [и др.]; под общ. ред. К.И. Микульский – М. : Наука, 2002. – 376 с.
- [4]. Шумпетер, Й.А. Теория экономического развития / Й.А. Шумпетер. – М. : Прогресс, 1982. – 455 с.
- [5]. Yager, R.A foundation for a theory of possibility / R.A. Yager // J. of Cybernetics. 1980. – Vol. 10, №.3. – P. 177-209.

DEVELOPMENT ONLINE PLATFORM EVALUATION AND FINANCING OF INNOVATIVE PROJECTS

V. F. ALEXSEEV, PhD
Associate professor of department of Design of Information and Computer Systems, Associate professor

D. V. LIKHACHEVSKY, PhD
Dean of the Faculty of Computer Design BSUIR, Associate professor

G. A. PISKUN, PhD
Associate professor of department of Design of Information and Computer Systems, Associate professor

Belarusian State University of Informatics and Radioelectronics, Republic of Belarus
E-mail: alexvikt.minsk@gmail.com, piskunbsuir@gmail.com, likhachevskyd@bsuir.by

Abstract. On the basis of modern methods, an approach to the design of an online platform is shown to evaluate the financing of innovation projects.

Key words: innovation, innovation project, B2B, UML, investor, innovator, information model.