

При нажатии на кнопку в главном окне программы «Соискатели» загружается окно «Соискатель», в котором располагается форма для заполнения сведений о клиенте (форма КПУ – карточки персонального учета граждан) (рис. 4). В данном окне можно завести все необходимые сведения о клиенте, который ищет работу, в соответствующие поля и добавить его в базу данных кадрового агентства. Также при нажатии кнопки в главном окне «Работодатели» открывается карточка для заведения сведений о работодателе (рис. 5).

Рис. 4. Окно «Соискатель»

Рис. 5. Окно «Работодатель»

Разработанная информационная система позволит уменьшить документооборот в организации и уменьшить время на обработку данных, увеличит скорость поиска необходимой информации, имеет удобный пользовательский интерфейс и выполняет все функции, необходимые для работы специалиста по работе с клиентами кадрового агентства.

МОБИЛЬНЫЙ МОНИТОРИНГ ТЕКУЩЕГО СОСТОЯНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

© А.А. Бурыкин, Ф.С. Луговой, В.В. Лавров, 2012
ФГАОУ ВПО «Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина», г. Екатеринбург

В настоящее все большую популярность набирают мобильные устройства широкого спектра применения (смартфоны) с унифицированными системами – Android (Google) и iOS (Apple). Такие устройства позволяют уже не только звонить, принимать вызовы, но и обладают большинством функций современных компьютеров. Можно смотреть фильмы,

играть в игры, слушать музыку, бороздить просторы Интернета и т.п. Помимо развлекательных функций, появляются и полезные приложения, например, офисные пакеты, позволяющие просматривать/редактировать популярные офисные форматы, а не так давно под мобильные платформы вышел известный графический пакет Adobe Photoshop [1]. Таким образом, несомненным конкурентным преимуществом любой информационной системы является наличие интерфейса с мобильного устройства. В статье рассматривается унифицированный интерфейс для мониторинга текущего состояния информационной системы.

При проектировании такой системы будем исходить от условия, что существуют множество объектов с информационными системами и конечные пользователи (лицо принимающее решение – ЛПР), которые будут получать данные из этих систем. Каждый пользователь может получать данные только из одной системы (объекта автоматизации), например, технолог доменного цеха получает данные только по своему цеху. Все данные хранятся на едином сервере (Сервер хранения данных), возможно, в облаке, поэтому на данном узле данные должны быть защищены от несанкционированного доступа, т.е. зашифрованы. Следовательно, информация шифруется перед передачей на сервер хранения данных. Архитектура данной системы показана на рис. 1.

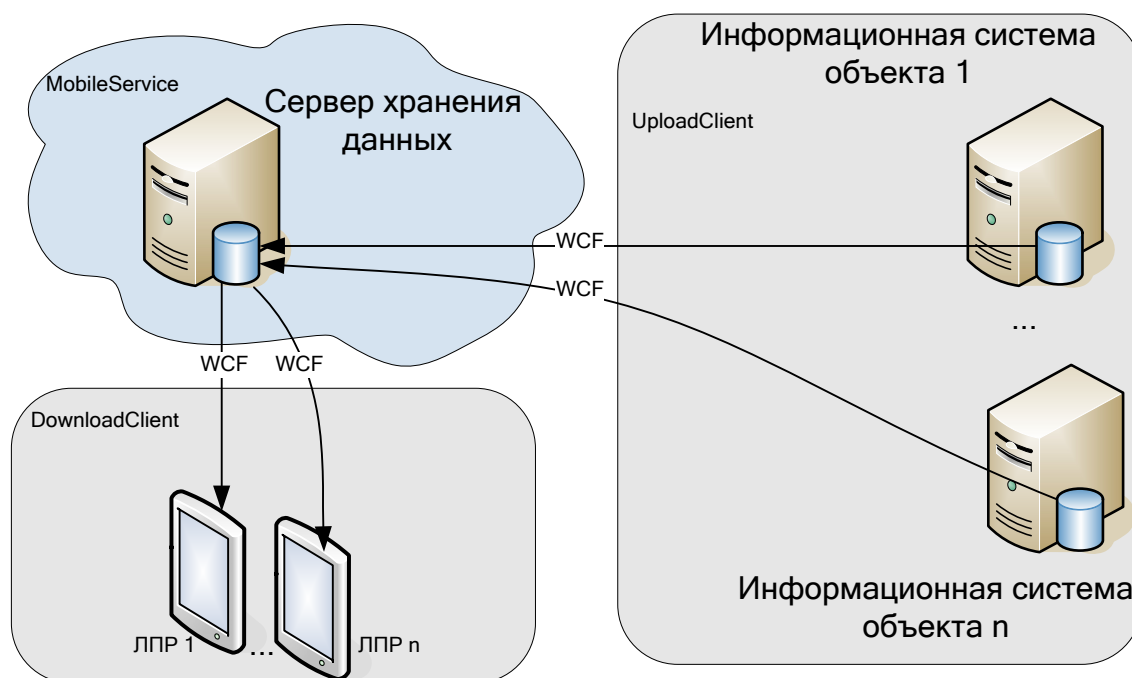


Рис. 1. Архитектура системы мобильного мониторинга

Для взаимодействия следует использовать платформу от компании Microsoft – Windows Communication Foundation (WCF) [2], которая позволяет передавать данные в SOAP [3] формате. Для платформы iOS есть уже наработки в этом направлении [4].

Пример запроса RequestForData:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <Action s:mustUnderstand="1"
      xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://tempuri.org/IService1/RequestForData</Action>
  </s:Header>
  <s:Body>
    <RequestForData xmlns="http://tempuri.org/">
      <inClient>0</inClient>
    </RequestForData>
  </s:Body>
</s:Envelope>
```

```

</RequestForData>
</s:Body>
</s:Envelope>

```

Пример ответа RequestForData:

```

<s:Envelope xmlns:s=«http://schemas.xmlsoap.org/soap/envelope/»>
  <s:Header />
  <s:Body>
    <RequestForDataResponse xmlns=«http://tempuri.org/»>
      <RequestForDataResult
        xmlns:a=«http://schemas.datacontract.org/2004/07/WcfServiceLibraryTest»
        xmlns:i=«http://www.w3.org/2001/XMLSchema-instance»>
        <a:Value>AAECAwQFBgcICQoLDA0ODxA=</a:Value>
      </RequestForDataResult>
    </RequestForDataResponse>
  </s:Body>
</s:Envelope>

```

В общем виде алгоритм работы всей системы выглядит следующим образом:

1. Информационная система объекта готовит данные для передачи, т.е. получает и шифрует информацию.
2. Информационная система объекта загружает данные на сервер хранения данных.
3. Сервер хранения данных помещает данные в БД, ставя в соответствие с объектом автоматизации.
4. ЛПР с мобильного устройства получает данные и расшифровывает их.

Следует отметить, что для работы системы требуется аутентификация/авторизация ЛПР на сервере хранения данных, для этого необходимо хранить информацию о пользователях (например, Login, PassHash, UserSid), привязанных к конкретному объекту автоматизации (рис. 2). В целях повышения безопасности UserSid может формироваться каждый раз при выполнении процедуры аутентификации.

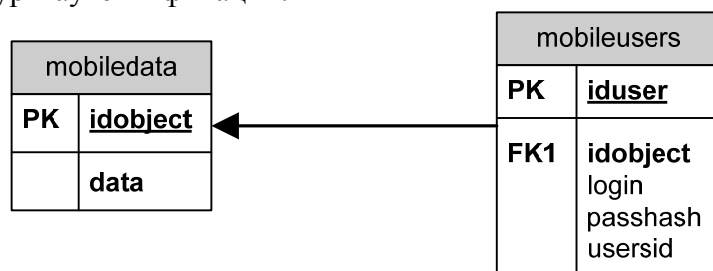


Рис. 2. Схема БД для сервера хранения данных

Объект автоматизации использует аналогичную процедуру аутентификации для загрузки данных в БД.

Рассмотрим теперь процедуру взаимодействия с ЛПР:

1. Процедура аутентификации, т.е. пользователь с мобильного устройства вводит логин/пароль и получает UserSid.
2. Далее пользователь с мобильного устройства:
 - a. При первом подключении вводит ключ для дешифровки (можно сканировать qr код).
 - b. Получает данные (*WrappedData*) с сервера хранения данных с помощью WCF вызова.
 - c. Расшифровывает данные и получает в результате XML объект типа *MobileData*.

Пример XML объекта MobileData.

```
<?xml version=«1.0»?>
```

```

<mobileData xmlns:xsi=«http://www.w3.org/2001/XMLSchema-instance»
xmlns:xsd=«http://www.w3.org/2001/XMLSchema»>
  <BranchId>0</BranchId>
  <Label>root</Label>
  <State>0</State>
  <Branches>
    <mobileData>
      <BranchId>1</BranchId>
      <Label>Выручка</Label>
      <State>0</State>
      <Branches>
        <mobileSkiData>
          <BranchId>10</BranchId>
          <Label>143,453 руб</Label>
          <State>1</State>
          <Branches />
        </mobileSkiData>
      </Branches>
    </mobileData>
    <mobileSkiData>
      <BranchId>2</BranchId>
      <Label>Оборудование</Label>
      <State>0</State>
      <Branches>
        <mobileData>
          <BranchId>20</BranchId>
          <Label>Турникеты</Label>
          <State>0</State>
          <Branches />
        </mobileData>
        <mobileData>
          <BranchId>3</BranchId>
          <Label>Проходы</Label>
          <State>0</State>
          <Branches>
            <mobileData>
              <BranchId>30</BranchId>
              <Label>436 проходов</Label>
              <State>1</State>
              <Branches />
            </mobileData>
          </Branches>
        </mobileData>
      </Branches>
    </mobileData>
  </mobileData>

```

d. На основе полученных данных строит дерево и показывает пользователю эти данные (*MobileData.Label*) в соответствии с состоянием. Например, если *MobileData.State* имеет значение 0, то зеленым, 1 – желтым, 2 – красным. При обновлении данных необходимо запоминать текущее положение пользователя (*MobileData.BranchId*), чтобы в случае добавления новых элементов остаться в текущем положении.

Процедура взаимодействия с объектом автоматизации выглядит следующим образом:

1. Процедура аутентификации, идентичная аутентификации пользователя.
2. Объект автоматизации с заданным интервалом времени вызывает метод на сервере хранения данных (с помощью WCF), передавая текущие данные. Таким образом, происходит обновление данных в БД в зашифрованном виде (ключ для расшифровки есть только у пользователя).

Поскольку реализация данной архитектуры такой системы (применение WCF) подразумевает использование платформы от компании Microsoft, то рассмотрим объектные сущности, которые будут использоваться для этого решения (рис. 3):

- Объект *MobileData*, содержит следующие поля:
 - *BranchId* – id отображаемого элемента интерфейса, *UInt32*
 - *Label* – текст на элементе интерфейса, *String*
 - *State* – состояние, необходимо, чтобы подкрашивать элемент интерфейса, *Byte*
 - *Branches* – список дочерних элементов, *List<MobileData>*
- Объект *WrappedData*, содержит, единственное поле:
 - *Value* – зашифрованный XML объект (*MobileData*), *Byte[]*

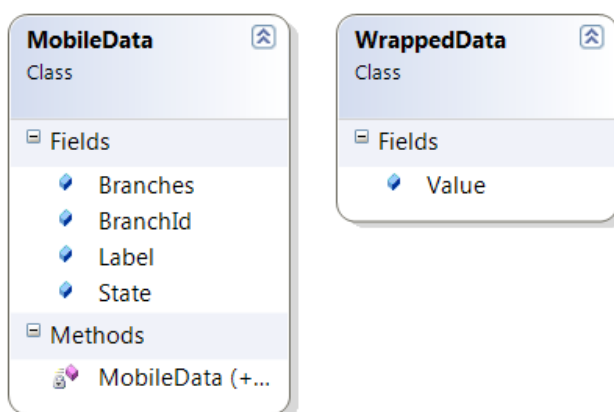


Рис. 3. Типы данных для системы мобильного мониторинга

Минимальный набор функций для системы мобильного мониторинга будет выглядеть следующим образом:

1. Аутентификация – передается связка логин/пароль, возвращается *UserSid*.
UInt32 Login(String inLogin, String inPwd);
2. Отправка данных – передается *UserSid* и экземпляр типа *WrappedData*, который содержит зашифрованный экземпляр типа *MobileData*.

void SendData(UInt32 inClient, WrappedData inData);

3. Запрос данных – передается *UserSid*, возвращается *WrappedData*.

WrappedData RequestForData(UInt32 inClient);

Предложенный вариант системы мобильного мониторинга позволяет абстрагироваться от бизнес логики конкретной системы автоматизации, поскольку интерфейс пользователя на мобильном устройстве строится динамически. Таким образом, это не только разработанный унифицированный программный интерфейс, но и в перспективе самостоятельная служба, не зависящая от автоматизированной системы объекта (системы учета рабочего времени, системы контроля доступа, платежно-пропускные системы и т.п.), которая позволяет осуществлять мониторинг текущего состояния информационной системы с мобильного устройства.

Список использованных источников

1. Компьютерра Блоги [web]. URL: <http://blogs.computerra.ru/25251> (дата обращения: 28.02.2012).

2. .NET Framework Developer Center [web]. URL: <http://msdn.microsoft.com/en-us/netframework/aa663324.aspx> (дата обращения: 28.02.2012).
3. SOAP. Материал из Википедии – свободной энциклопедии [web]. URL: <http://ru.wikipedia.org/wiki/SOAP> (дата обращения: 28.02.2012).
4. iPhone SDK Development Forums [web]. URL: <http://www.iphonedevsdk.com/forum/iphone-sdk-development/39819-how-call-wcf-service.html> (дата обращения: 28.02.2012).

СИСТЕМА УПРАВЛЕНИЯ ПРОЕКТНЫМИ ЗАДАЧАМИ JIRA

© А.Г. Быков, А.И. Веселов, 2012
ООО «МИРАЛОДЖИК ИТ», г. Екатеринбург

Введение. Atlassian JIRA – это основанное на web средство для управления проектами и задачами. JIRA может применяться во всех случаях, когда необходимо организовать работу сотрудников, эффективно назначать им задачи, иметь мгновенные средства контроля их выполнения. Система разрабатывается компанией Atlassian Software Systems и претендует на звание стандарта де-факто в этой области. Система успешно используется в следующих известных организациях: Росгосстрах, «МТТ», «Ренесанс страхование», «Финансовая корпорация „Уралсиб“», «Альфа-страхование», «КИТ-Финанс» и многих других.

Наиболее эффективно система служит для управления задачами, способ решения которых не определяется регламентом подразделения во всех деталях. Любая деятельность, в которую входит синтез, анализ, постановка целей, выбор средств достижения, поиск и обработка информации, оценка полученных результатов может успешно вестись в системе JIRA. Вышеперечисленная деятельность называется **проектной**. Любой руководитель холдинга и многие специалисты занимаются проектной деятельностью. Выбранная система позволяет существенно увеличить эффективность работы сотрудников предприятия.

Представленная система планирования является мощным инструментом руководителя, а также предоставляет удобный функционал для рядового персонала. Кроме того, система JIRA, интегрируется с базой знаний компании Atlassian, что позволяет использовать последнюю как хранилище информации, связанной с выполнением задач. Работа с базой знаний рассматривается в отдельном документе.

Ключевые понятия системы. Ключевыми понятиями в JIRA являются Проекты и Задачи. Основные настройки Проекта определяются в Схемах. Схемы определяют права доступа к Задачам, правила уведомлений об изменении атрибутов Задач, уровни безопасности. В Проекте определяются основные Схемы. Несколько Проектов можно **визуально** объединить в Группу Проектов (никакими другими настройками Группы Проектов не обладают).





4.1 Группа разработки сервисов (нач. Быков А.)		
Проект	Код	Руководитель проекта
 4.1 Группа разработки сервисов	GRS	Быков Александр Георгиевич
 4.1.1 Отдел РазВития Сервисов	WNT	Аксёнов Артём Анатольевич
 4.1.2 Отдел РазРаботки Сервисов	WBG	Скробов Даниил Леонидович
 4.1.3 Отдел Технологий Абонентских Сервисов	CS	Космаков Алексей Евгеньевич
 4.1.4 Отдел дизайна	DSGN	Клевакин Денис Анатольевич

Рис. 1. Группа Проектов «4.1 Группа разработки сервисов»