



ИНСТИТУТ ЗА МАТЕМАТИКУ И ИНФОРМАТИКУ
ПРИРОДНО-МАТЕМАТИЧКИ ФАКУЛТЕТ
УНИВЕРЗИТЕТ У КРАГУЈЕВЦУ

ДИПЛОМСКИ РАД

Синхронизација календара за *ownCloud* платформу

Студент:
Владимир Варагић

Професор:
др Милош Ивановић

Септембар 2016.

Садржај

1	Увод	3
2	Преглед коришћених технологија и радно окружење	4
3	<i>OwnCloud</i> пројекат	5
3.1	<i>Calendar</i> подапликација	6
4	Синхронизација календара за <i>ownCloud</i> платформу	8
4.1	Жељене функционалности	8
4.1.1	Аутентификација	9
4.1.2	Синхронизација догађаја на захтев	10
4.1.3	Аутоматска синхронизација догађаја	11
4.1.4	Преглед преузетих догађаја	12
4.1.5	Приказ нотификација	12
4.2	Идеје за даљи развој	14

Листа скраћеница

- *API - Application Programming Interface,*
- *BSD - Berkeley Software Distribution,*
- *CalDAV - Calendaring Extensions to WebDAV,*
- *GPL - GNU Public Licence,*
- *IT - Information Technology,*
- *MIT - Massachusetts Institute of Technology,*
- *SVN - Subversion,*
- *URL - Uniform Resource Locator.*

Глава 1

Увод

Технолошки развој, а посебно развој интернета, је довео до тога да је интернет постао саставни и готово неизоставни део свакодневног живота, а постојање и широка употреба мобилних уређаја (паметних телефона, нетбук рачунара, таблет рачунара,...) временом је развила потребу за сталним приступом приватним подацима и документима. Самим тим складиштење приватних података и докумената на кућним стоним рачунарима полако постаје превазиђено. Као алтернатива намеће се рачунарство у облаку.

Коришћењем рачунарства у облаку могуће је складиштити личне податке на приватном удаљеном серверу, при том имајући могућност приступа тим подацима са било које локације на интернету, употребом било ког мобилног уређаја, што се у великој мери преклапа са наведеним тенденцијама савременог друштва. Поред великог броја комерцијалних решења, попут *Dropbox-a*, развијена су и многобројна "отворена" решења која корисницима на једноставан и интуитиван начин обезбеђују већу контролу над подацима. Једно од таквих "отворених" решења је и *OwnCloud*.

Поред могућности складиштења приватних података, *OwnCloud* нуди и могућност вођења календара активности, односно неке врсте е-планера. Развој десктоп клијента који би имао функцију подсетника, а који би садржај наведеног календара активности користио као извор података, је тема овог рада. У наставку ће бити укратко описан садржај поглавља овог рада.

Поглавље *Преглед коришћених технологија и радно окружење* представља опис технологија које су коришћене приликом развоја решења које је тема овог рада.

Поглавље *ownCloud* укратко описује пројекат и апликацију чије сервисе дати десктоп клијент треба да користи. Опис десктоп клијента и приказ кључних делова програмског кода биће представљен у поглављу *ownCloudCalendar*.

Глава 2

Преглед коришћених технологија и радно окружење

Развој софтверских решења се генерално може поделити на развој комерцијалних софтвера, који се штите власничким лиценцама, и на развој софтвера са нешто "отворенијим" лиценцама, док је програмски код обично отворен и доступан. Када се говори о отвореним лиценцама мора се бити обазрив према отворености и слободи коју лиценца као таква пружа. Примера има пуно, а познате су BSD, GPL и MIT[1] лиценце, које се користе у свету софтвера отвореног кода.

Постоји неколико различитих алата за контролу изворног кода. До скоро је *Subversion* био најзаступљенији, али у последње време *Git* преузима примат на тржишту, јер је заснован на другачијим принципима, тако да више задовољава потребе корисника. Као такав био је погодан за коришћење и у овом раду заједно са слободним и бесплатним *Git* репозиторијумом *GitHub*, који поред простора који пружа, даје и неопходну статистику везану за број учесника на пројекту, њихову активност итд.

Један од водећих изазова у развоју десктоп апликација је био у томе да се нађе начин за превазилажење ограничења која су изазавана оперативним системима на којима те апликације треба да раде. Разлике у концептима и техничким специфичностима које постоје међу водећим оперативним системима утицале су на то да десктоп апликације развијене за једним оперативним систем не могу да раде на осталим оперативним системима без одговарајућег прилагођавања. Како би се ова ограничења превазишла, јавила се потреба за развојем платформи које ће омогућити да десктоп апликације без проблема раде на свим оперативним системима. Једна од таквих платформи је и *XWT*[2].

XWT је *.NET* мултиплатформски алат за развој корисничког интерфејса. Омогућава развој десктоп апликација које могу да раде на различитим платформама без потребе да се код прилагођава свакој од њих. Разлика у односу на традиционални приступ у развоју десктоп апликација је у томе што се контроле исцртавају динамички, у самом коду, а сам *XWT API* има способност да у зависности од платформе изабере одговарајуће контроле.

У развоју решења, које је тема датог рада, коришћене су и следеће готове компоненте са отвореним лиценцама:

- *DDay.iCal*[3] - библиотека класа за рад са календаром за окружење *.NET-a 2.0* и новије верзије,
- *CalDAV*[4] - протокол за синхронизацију календара. Представља интернет стандард који омогућава клијенту да приступи информацијама о планираним догађајима на удаљеном серверу. Дозвољава истовремени приступ истим информацијама од стране више клијената, чиме се омогућава кооперативно планирање и дељење информација.

Глава 3

OwnCloud пројекат

Рачунарство у облаку представља скуп ресурса, чији је задатак да омогући складиштење велике количине података или извршавање великог броја процеса. Основна карактеристика рачунарства у облаку је та да корисницима омогућава коришћење удаљених ресурса, при чему им није дозвољен физички приступ датим ресурсима. Пораст броја корисника са оваквим захтевима утицао је и на појаву великог броја комерцијалних платформи које нуде услугу рачунаарства у облаку, међу којима су *Amazon*, *Microsoft*, *Dropbox* и многе друге. Једно од таквих решења је и *ownCloud*[5] пројекат.

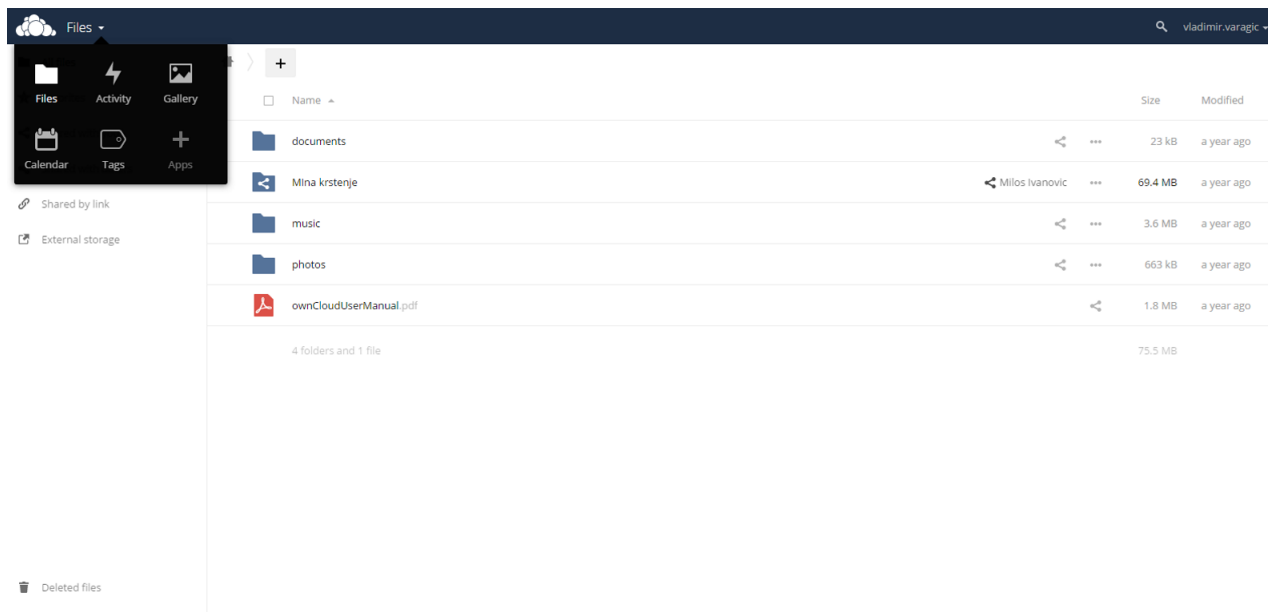
На почетку основна идеја *OwnCloud* пројекта је била да се обичном кориснику омогући да има приватно складиште на којем ће моћи да складишти своје податке. У међувремену, овај пројекат је добио много унапређења која нису директно везана за само складиштење података. Творац пројекта Франк Карличек је идеју о потреби решења са "отвореним" лиценцама изнео на скупу програмера и успео је да обезбеди неопходан број учесника који ће допринети развоју и популаризацији овог пројекта.



Слика 3.1: *OwnCloud* лого

Постоје три могућности за приступ подацима на *ownCloud-u*:

- Десктоп апликација - омогућава кориснику да складишти и/или преузима податке са удаљених ресурса,
- *WebDAV* технологија - погоднија од десктоп апликације, која се мора инсталирати на сваком рачунару, али са друге стране ограничава корисницима приступ само до података,
- Веб апликација - нема ограничења као друге две опције. Дакле, омогућен је приступ свим погодностима које *ownCloud* портал нуди и које ће бити представљене у наставку.

Слика 3.2: Кориснички интерфејс *ownCloud* веб апликације

Као што се може видети на Слици 3.2, неке од основних функционалности су:

- Приказ листе фајлова и директоријума тренутно пријављеног корисника,
- Могућност додавања нових садржаја,
- Могућност брзе претраге садржаја,
- Листа доступних апликација,
- Могућност приступа приватним подацима корисника, као и могућност одјаве.

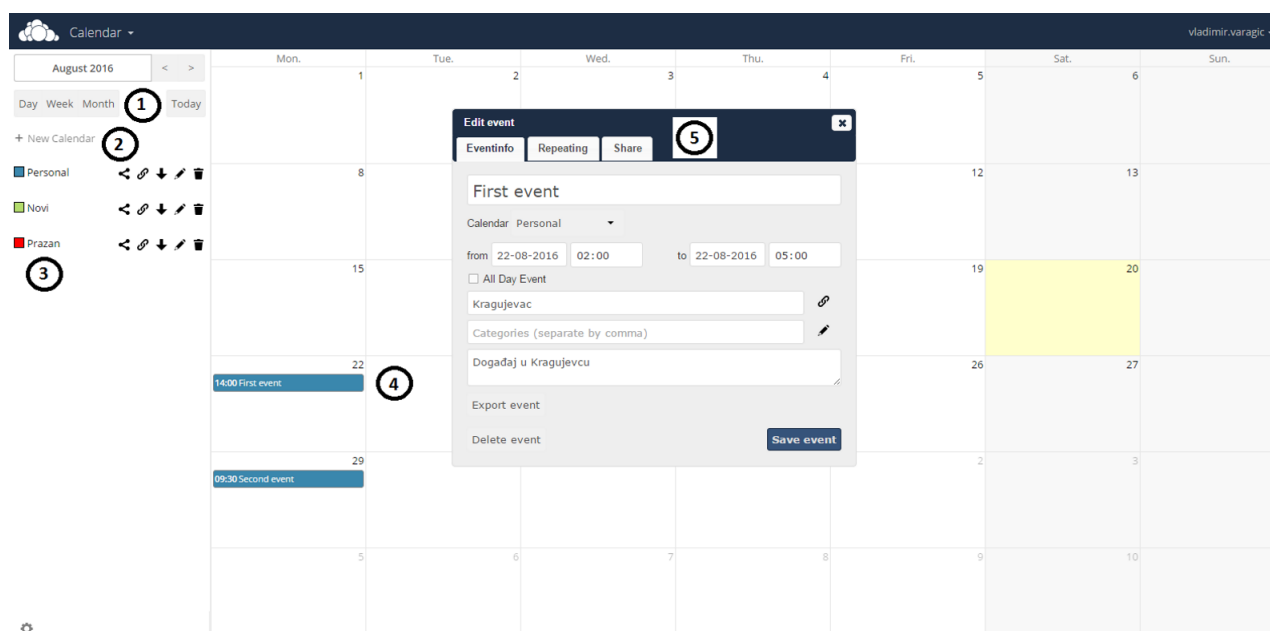
Поред могућности складиштења и приступа подацима *ownCloud* веб апликација нуди могућност коришћења њених подапликација:

- Вођење листе контаката,
- Делење корисничких података између корисника истог складишта,
- Праћење активности корисника,
- Листа доступних апликација,
- Праћење календара, итд.

Посебна пажња биће посвећена *Calendar* подапликацији.

3.1 *Calendar* подапликација

Као што је већ раније поменуто, основна идеја и функција *ownCloud* апликације је била да омогући складиштење података корисника, што није било довољно атрактивно. Увођењем платформе за креирање нових сервиса (подапликација) овај проблем је превазиђен. Један од њих је и *Calendar* подапликација.



Слика 3.3: Кориснички интерфејс *ownCloud Calendar* подапликације

На Слици 3.3 су приказане све њене функционалности:

1. Начин приказа календара (дневни, недељни, месечни),
2. Креирање новог календара,
3. Приказ постојећих календара,
4. Приказ постојећих догађаја,
5. Администрација (унос, измена, брисање) догађаја.

Подапликација *Calendar* је прилично једноставна и интуитивна за коришћење.

Глава 4

Синхронизација календара за *ownCloud* платформу

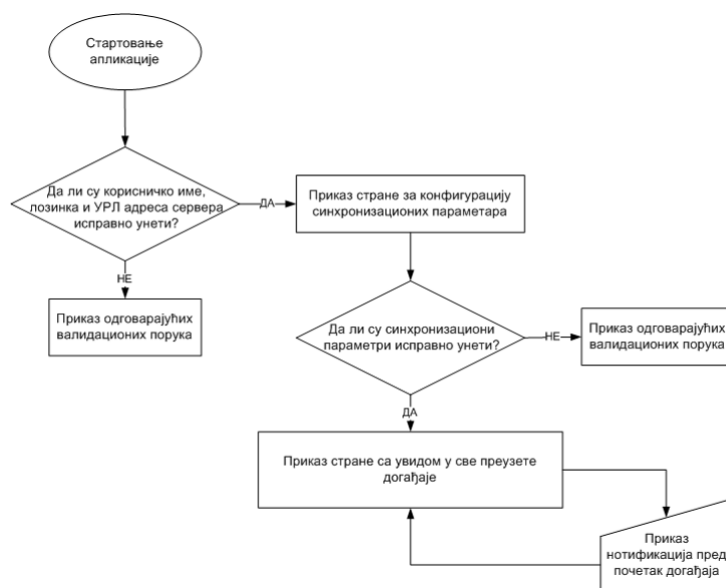
У претходним поглављима описани су основни концепти технологија и окружења који су коришћени у развоју датог пројекта, са циљем да се читаоцу омогући да формира слику комплетног, заокруженог, решења. Сам пројекат, који је тема овог рада, може се посматрати као део тог решења. У овом поглављу фокус ће бити постављен на појашњења неких делова његове имплементације.

4.1 Жељене функционалности

Актуелна, званична, верзија *ownCloud* десктоп клијента обезбеђује само синхронизацију докумената који се налазе на *ownCloud* платформи. Основни циљ овог пројекта јесте да се развије решење, у виду мултиплатформског десктоп клијента, које би омогућило преузимање информација о креираним догађајима на *ownCloud* календару и приказ одговарајућих обавештења. Апликација има следећи скуп функционалности:

- синхронизација догађаја на захтев,
- аутоматска синхронизација догађаја,
- могућност управљања аутоматском синхронизацијом (потребна/није потребна, дефинисање временског интервала након којег ће се стартовати,...),
- преглед преузетих догађаја,
- приступ делу за администрацију догађаја на веб порталу *ownCloud* платформе,
- приказ одговарајућег обавештења, непосредно пре почетка неког догађаја.

Ток активности које треба да обезбеде ове функционалности описан је на дијаграму 4.1.

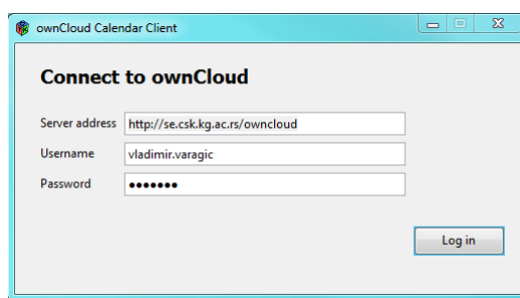


Слика 4.1: Дијаграм тока активности

На основу приказаног алгорита може се стећи јасна и потпуна слика о начину рада саме апликације. У наставку ће бити детаљније објашњене неке интересантније функционалности и биће приказани делови програмског кода, док се комплетан код пројекта може погледати на одговарајућем репозиторијуму[6].

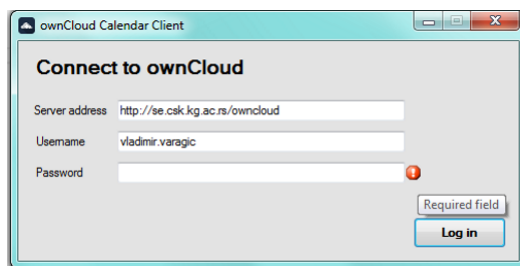
4.1.1 Аутентификација

Аутентификација корисника на веб портал *ownCloud* платформе одрађена је коришћењем класа *WebClient*, *NetworkCredential* које су саставни део *.NET Framework-a*. Подаци унети на форми за пријаву на систем (Слика 4.2), која се приказује након стартовања апликације, се прослеђују на верификацију.



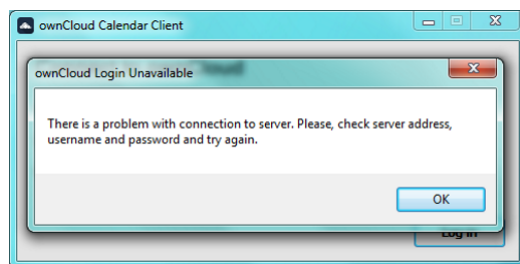
Слика 4.2: Форма за пријаву на систем

Сви подаци на форми за пријаву су обавезни, па се у случају да неки податак није унет, прикаже одговарајући индикатор (Слика 4.3).



Слика 4.3: Форма за пријаву на систем

Такође, у случају да неки од података који се уносе приликом пријаве на апликацију (адреса сервера, корисничко име или лозинка) није исправан приказује се одговарајућа порука (Слика 4.4).

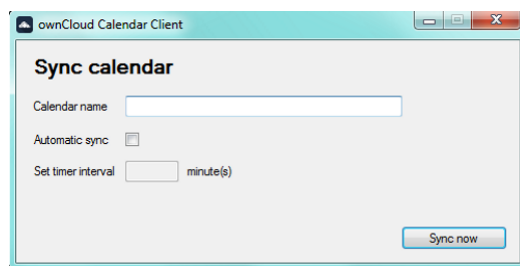


Слика 4.4: Форма за пријаву на систем

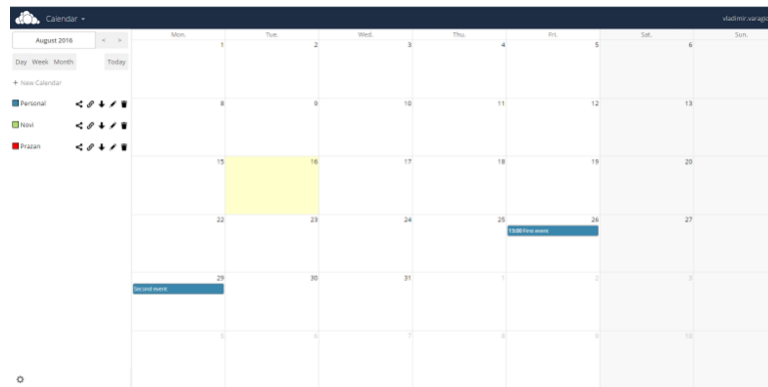
У супротном, ако су сви подаци исправни, корисник успешно приступа апликацији и приказује му се форма за синхронизацију догађаја са *ownCloud* календара.

4.1.2 Синхронизација догађаја на захтев

Као што је већ наведено у поглављу 4.1.1 *Ауентификација*, након успешног приступа апликацији кориснику се приказује форма за конфигурацију синхронизације (Слика 4.5).

Слика 4.5: Синхронизација догађаја са *ownCloud* календара

OwnCloud платформа омогућава кориснику да на порталу води више различитих календара тј. да календар дели у различите категорије.

Слика 4.6: *OwnCloud* календар

Са друге стране, синхронизацијом се у једном тренутку могу преузети само догађаји који су везани за једну категорију, тако да је назив календара обавезан податак приликом синхронизације. Такође, приликом покретања синхронизације ради се валидација исправности назива календара и уколико не постоји календар са унетим називом кориснику се прикаже одговарајућа порука. У супротном, ако је унет исправан назив календара, кориснику се приказују догађаји који постоје на наведеном календару. Сам приказ података о догађају биће детаљно описан у секцији 4.1.4 *Преглед перузетих догађаја*.

Методe којима се синхронизују подаци приказани су на слици 4.7.

```

180 private void btnSyncCalendar_Click(object sender, EventArgs e)
181 {
182     try
183     {
184         if (ValidateControls())
185         {
186             ICalendarCollection iCalCollection = GetCalendarEventsData();
187
188             if (iCalCollection == null)
189             {
190                 ShowMessage("There is no calendar with the name " + txtCalendarName.Text.Trim(), "ownCloud Calendar Sync Unavailable");
191             }
192             else
193             {
194                 Hide();
195                 int? syncTimeInterval = null;
196                 if (!String.IsNullOrEmpty(txtTimeInterval.Text))
197                 {
198                     syncTimeInterval = Convert.ToInt32(txtTimeInterval.Text);
199                 }
200                 EventsList eventsList = new EventsList(iCalCollection, cbAutomaticSync.Checked, syncTimeInterval, txtCalendarName.Text, serverUrl, username, password, serverAddress);
201                 eventsList.ShowDialog();
202                 if (eventsList.IsHidden)
203                 {
204                     HideForm();
205                 }
206                 else
207                 {
208                     Show();
209                 }
210             }
211         }
212     }
213     catch (Exception ex)
214     {
215         HandleException(ex);
216     }
217 }
218
219 private ICalendarCollection GetCalendarEventsData()
220 {
221     ownCloudCalendarConnector connector = new ownCloudCalendarConnector();
222
223     string url = serverAddress + cCalDavUrlExtension + username + cCalDavUrlExtensionSlash + txtCalendarName.Text.Trim().ToLower() + cCalDavUrlExtensionExport;
224     serverUrl = new Uri(url);
225
226     return connector.ownCloudCalendar_GetEvents(serverUrl, username, password);
227 }

```

Слика 4.7: Методе за синхронизацију догађаја са *ownCloud* календара

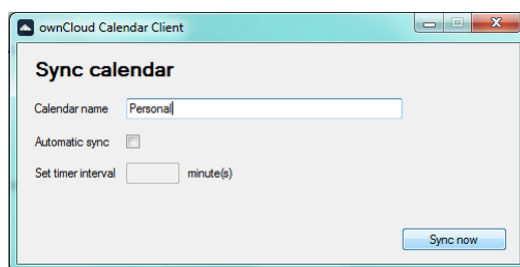
4.1.3 Аутоматска синхронизација догађаја

Поред наведене функционалности за синхронизацију догађаја на захтев, омогућена је и функционалност аутоматске синхронизације догађаја. Уколико корисник жели да користи дату функционалност, потребно је да чекира опцију *Automatic sync* на форми за син-

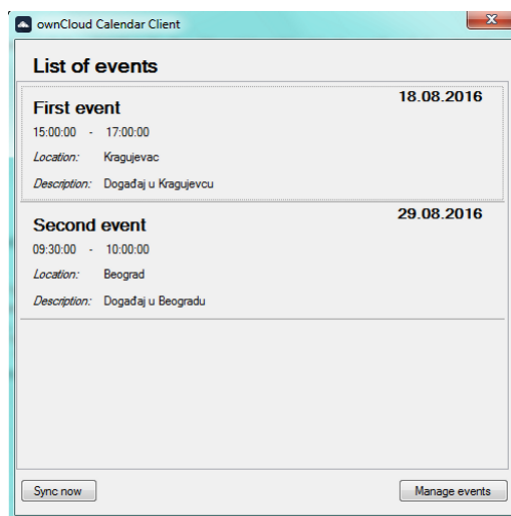
хронизацију догађаја (Слика 4.5). Када је опција *Automatic sync* чекирана, податак *Sync time interval* је обавезан. Дакле, након дефинисања наведених података, апликација ће аутоматски синхронизовати догађаје са одговарајућег календара у наведеном временском интервалу. Времески интервал се дефинише у минутима и одговарајућом валидацијом онемогућено је да вредност овог податка буде било шта што није цео позитиван број.

4.1.4 Преглед преузетих догађаја

Када су сви обавезни подаци исправно унети, било да је у питању синхронизација догађаја на захтев, било да је у питању аутоматска синхронизација догађаја, кликом на дугме *Sync now* (Слика 4.8) пружају се догађаји са одговарајућег календара и прикаже се форма са листом догађаја (Слика 4.9).



Слика 4.8: Синхронизација догађаја са *ownCloud* календара



Слика 4.9: Приказ преузетих догађаја

Са форме за Преглед преузетих догађаја (Слика 4.9) корисник у сваком тренутку може поново да покрене синхронизацију догађаја (кликом на дугме *Sync now*), без обзира на то да ли је функционалност аутоматске синхронизације изабрана или не. Корисник, такође, има могућност да са форме за Преглед преузетих догађаја (Слика 4.9), кликом на дугме *Manage events* приступи календару на веб порталу *ownCloud* платформе (Слика 4.6) и администрира (креира нове, ажурира постојеће, брише) догађаје.

4.1.5 Приказ нотификација

Поред описаних функционалности апликација има још једну, вероватно најзанимљивију функционалност, а то је приказ одговарајућих нотификација у вези са преузетим догађајима. Нотификације се приказују према унапред дефинисаним параметрима:

- први параметар (*notificationMessageTimerInMinutes*) представља временски период (у минутима) којим се дефинише колико минута пре стартовања догађаја приказати одговарајућу нотификацију,
- други параметар (*notificationPingTimeInterval*) представља временски период (у милисекундама) којим се дефинише колико често ће се проверавати да ли је први параметар достигао дефинисану вредност.

Ови параметри су конфигурабилни и део су конфигурационог фајла (Слика 4.10).

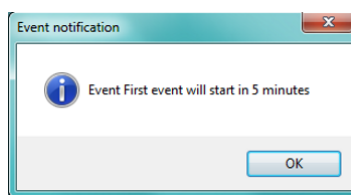
```

1 <?xml version="1.0" encoding="utf-8"?>
2 <configuration>
3   <appSettings>
4     <add key="notificationPingTimeInterval" value="60000"/>
5     <add key="notificationMessageTimerInMinutes" value="5"/>
6   </appSettings>
7   <startup>
8     <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
9   </startup>
10  <runtime>
11    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
12      <dependentAssembly>
13        <assemblyIdentity name="System.Web.Mvc" publicKeyToken="31bf3856ad364e35" culture="neutral" />
14        <bindingRedirect oldVersion="0.0.0.0-5.1.0.0" newVersion="5.1.0.0" />
15      </dependentAssembly>
16    </assemblyBinding>
17  </runtime>
18 </configuration>

```

Слика 4.10: Конфигурациони фајл

Дакле, у складу са дефинисаним вредностима наведених параметара, апликација сваког минута проверава да ли постоји догађај који ће стартовати за 5 минута и у случају да такав догађај постоји, кориснику се прикаже одговарајућа нотификација (Слика 4.11).



Слика 4.11: Приказ нотификације

У наставку је приказана метода којом је дата функционалност имплементирана (Слика 4.12).

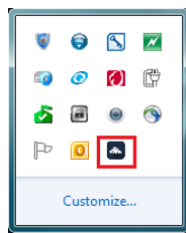
```

277
278 private void SetNotificationTimer()
279 {
280     Timer timer = new Timer();
281     int pingTimeInterval;
282
283     pingTimeInterval = Convert.ToInt32(ConfigurationManager.AppSettings["notificationPingTimeInterval"].ToString());
284     timer.Tick += new EventHandler(CheckEventStartTime);
285
286     timer.Interval = pingTimeInterval;
287     timer.Start();
288     CheckEventStartTime(this, null);
289 }
290

```

Слика 4.12: Метода за приказ нотификација

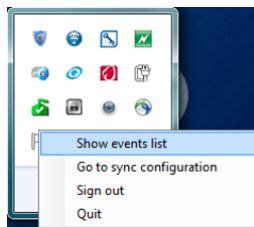
Поред функционалности описаних у претходним секцијама, споменућемо још нека својства апликације. Најпре, треба нагласити да је апликација *Single instance*, односно у једном тренутку могуће је покренути само једну инстанцу апликације. У случају да корисник покуша да покрене више инстанци апликације, то му неће бити дозвољено и приказаће се одговарајућа порука. Такође, требало би напоменути да се кликом на дугме *Close* на форми за Приказ преузетих догађаја апликација не затвара, већ се само минимизује, тј. апликација је и даље покренута и иконица апликације налази се у таскбару (Слика 4.13).



Слика 4.13: Таскбар - приказ иконице

Десни клик на иконицу у таскбару нуди следеће опције (Слика 4.14):

- отварање форме за приказ преузетих догађаја,
- отварање форме за унос параметара синхронизације,
- одјава са апликације и приказ форме за пријаву,
- затварање апликације.



Слика 4.14: Додатне опције

4.2 Идеје за даљи развој

Иако је функционално исправна, постојећу верзију апликације треба посматрати само као полазни корак у развоју коначног производа. Актуелна верзија апликације има својеврсна ограничења условљена коришћеним API-има (нпр. немогућност синхронизације више календара истовремено). Унапређења датих API-а или појава нових утицали би на то да се појави потреба за имплементацијом додатних функционалности. Са друге стране, постојећа верзија се такође може унапредити на више начина:

- побољшање корисничког интерфејса,
- предефинисани прикази догађаја (за разлику од актуелног приказа свих догађаја),
- ...

Библиографија

- [1] BSD, GPL и MIT лиценце, <http://producingoss.com/en/license-choosing.html>
- [2] XWT платформа, <https://github.com/mono/xwt/tree/master/Xwt/Xwt>
- [3] DDay.iCal - an iCalendar class library, <https://sourceforge.net/p/dday-ical/wiki/Home/>
- [4] CalDav, <http://caldav.calconnect.org/index.html>
- [5] Званична страница *OwnCloud* пројекта, <http://owncloud.org/>
- [6] Репозиторијум *ownCloud Calendar Synchronization* апликације, <https://own-cloud-calendar.googlecode.com/svn>