

۴-۱- نیازمندی‌های سیستم

در این بخش قصد داریم نیازمندی‌های سیستم را مشخص کنیم. این نیازمندی‌ها با مراجعه به کتب و مقالات مربوط به ریزهسته‌ها و سیستم‌های عامل جمع آوری شده‌اند؛ در این پروژه برای کاهش ریسک‌ها، نیازمندی‌ها را تا حد امکان ساده در نظر گرفته‌ایم. البته دقت شود که ساده در نظر گرفتن نیازمندی‌ها به این معنی نمی‌باشد که سیستم طراحی شده یک ریزهسته نسل سوم نمی‌باشد، چرا که اصلی‌ترین ویژگی ریزهسته‌های نسل سوم صحت سنجی آن‌ها بصورت صوری می‌باشد و ما هم در این پروژه سیستم خود را بصورت صوری و با استفاده از متد B صحت سنجی کرده‌ایم.

۴-۱-۱- نیازمندی‌ها

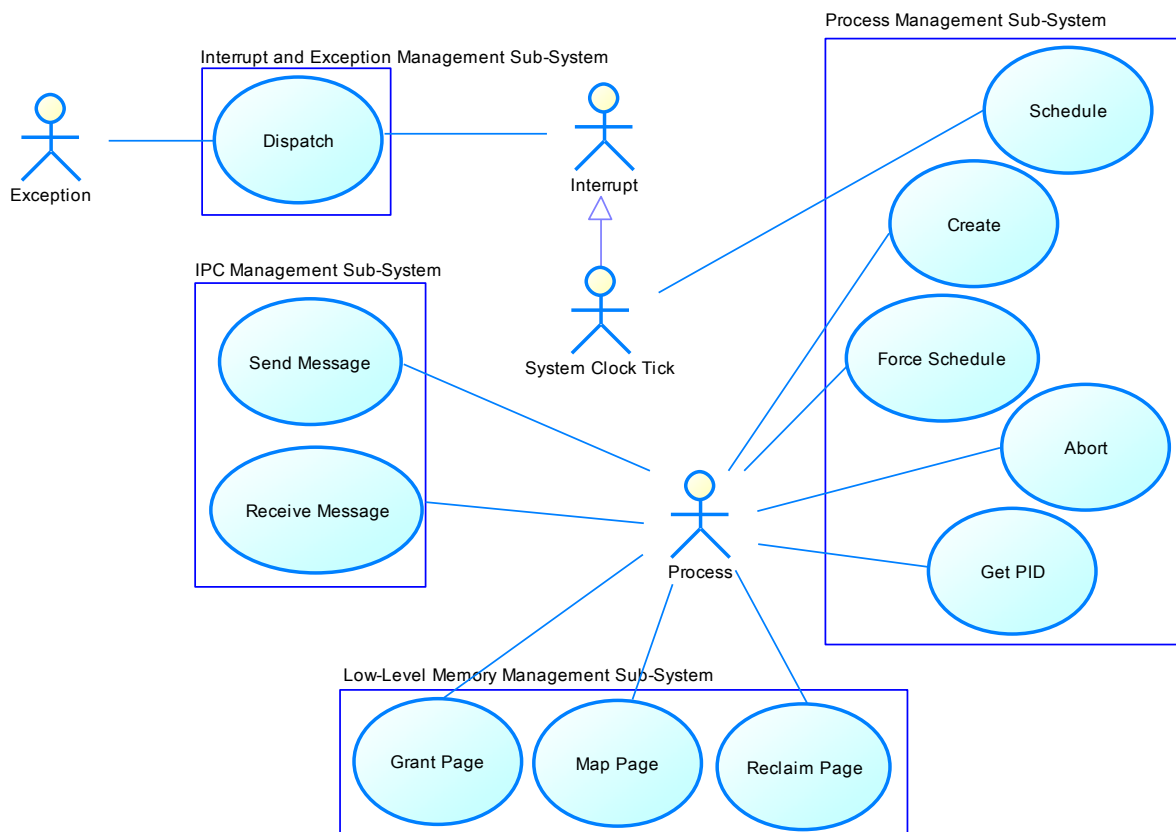
در افراطی‌ترین حالت، [LIED95] می‌گوید که ریزهسته‌ها می‌بایست حداقل سطوح تجرید را برای برنامه‌های کاربردی فراهم آورند؛ این سطوح تجرید نخها و فضاها را آدرس می‌باشند. در این پروژه برای کاهش ریسک، فرض می‌کنیم هر فضای آدرس مختص فرآیندی است که تنها یک نخ در آن در حال اجرا شدن می‌باشد. لازم به ذکر است که این کاهش حوزه پروژه با سند پیشنهاد پروژه مغایرت ندارد، چرا که در این سند اعلام کرده‌ایم که سیستم عامل ما قادر به زمانبندی خواهد بود و از آنجایی که زمانبندی حاصل از کاهش حوزه از قدرت سیستم نمی‌کاهد مجاز به این کار هستیم.

ریزهسته‌های نسل سوم بنا به [KLEI09] می‌بایست بصورت صوری صحت سنجی شده و همچنین دارای مکانیزم‌های امنیتی باشند. صحت سنجی صوری می‌تواند یک ریزهسته را در زمره ریزهسته‌های نسل سوم قرار دهد؛ باز در اینجا یک کاهش حوزه صورت گرفته است و آن هم جاسازی مکانیزم‌های امنیتی در ریزهسته می‌باشد. باز به منظور کاهش ریسک پروژه ریزهسته را فاقد مکانیزم‌های امنیتی در نظر می‌گیریم. لازم است بدانید که این کاهش حوزه با تعریف موجود در سند پیشنهاد مغایرت ندارد چرا که وجود صحت سنجی صوری موجب تولید یک ریزهسته نسل سوم خواهد شد.

در ادامه بیشتر در مورد نیازمندی‌های سیستم بحث و آنها را بصورت دقیق‌تر بیان خواهیم کرد.

۴-۱-۲- موارد استفاده

در این بخش بر آنیم که نیازمندی‌های سیستم را بصورت دقیق‌تر با استفاده از ابزارهای زبان UML بیان کنیم. همانطور که گفته شد هر ریزهسته‌ای حداقل باید سطح تجرید فرآیند^۱ و فضای آدرس را تامین کند. تامین هر یک از این سطوح تجرید سیستم عامل را ملزم می‌کند که فراخوان‌های سیستم خاص این سطوح تجرید را نیز تامین کند. برای این منظور ما فراخوان‌هایی را برای ایجاد، خاتمه، بلوک کردن فرآیندها در نظر گرفته‌ایم؛ همچنین فراخوان‌هایی را برای اهدا، نگاشت، و بازپس‌گیری بخش‌هایی از فضای آدرس، و نیز فراخوان‌هایی برای ارسال و دریافت پیام‌ها بین فرآیندها در نظر گرفته‌ایم. سیستم باید قادر باشد فرآیندها را زمانبندی کند برای این منظور از وفقه تیک ساعت سیستم استفاده خواهیم کرد، همچنین ریزهسته بودن، سیستم را ملزم به تبدیل وقفه‌ها و استثنائات به پیام‌های بین فرآیندی می‌کند. در شکل ۱ دیاگرام Use-Case سیستم را مشاهده می‌کنید که این نیازمندی‌ها را روشن‌تر می‌کند.



شکل ۱- نمودار موارد استفاده ریزهسته در حال توسعه.

^۱ البته این سطح تجرید در اصل نخ می‌باشد، ولی در این پروژه از فرآیند به جای فرآیند و نخ استفاده خواهیم کرد.

در این مرحله می‌بایست تک تک این موارد استفاده را شرح دهیم و سناریوهای استفاده را بیان کنیم.

۴-۱-۳- شرح موارد استفاده

Use-Case	Schedule	
Primary Actor	System Timer Tick	
Description	این مورد استفاده موجب فعال شدن زمانبند سیستم شده و فرآیند بعدی را که بیش از همه فرآیندها دارای شایستگی اجرا شدن است را به حالت در حال اجرا در می‌آورد. در صورتی که صف فرآیندهای آماده اجرا و نیز صف فرآیندهای در حال اجرا خالی باشد، سیستم متوقف می‌شود.	
Pre-condition	فرض می‌شود که صف فرآیندهای در حال اجرا و صف فرآیندهای آماده اجرا هر دو با هم خالی نیستند.	
Trigger	این مورد استفاده با هر وقفه تیک سیستم، و نیز بعد از هر فراخوان سیستمی فعال می‌شود.	
Typical Course of Events	Actor Action	System Response
	گام (۱): ساعت سیستم وقفه را ایجاد می‌کند؛	گام (۲): زمانبند سیستم فعال می‌شود؛ گام (۳): صف فرآیندهای جاری (در حال اجرا) را به انتهای صف فرآیندهای آماده اجرا اضافه می‌کند؛ گام (۴): فرآیندی که در سر صف فرآیندهای آماده اجرا قرار دارد را به عنوان فرآیند جاری انتخاب می‌کند؛ گام (۵): با خروج از زمانبند فرآیند جاری به اجرا در می‌آید.
Alternative Course:		
Post-condition	بعد از خروج از زمانبند، فرآیندی که بیش از همه فرآیندهای دیگر شایستگی اجرا شدن را دارد به عنوان فرآیند جاری برای اجرا شدن انتخاب می‌شود.	
Business Rules	<ul style="list-style-type: none"> سیاست زمانبندی Round-Robin ساده و بدون اولویت است، بنابراین فرآیندی بیش از همه شایستگی اجرا شدن را دارد، همواره که در سر صف برنامه‌های آماده قرار گرفته است؛ از آنجایی که این سیستم عامل برای یک سیستم تک پردازنده‌ای طراحی 	

	<p>می‌شود، صف فرآیندهای در حال اجرا (جاری) همواره صفی است با اندازه یک؛</p> <ul style="list-style-type: none"> در صورتی که صف فرآیندهای آماده اجرا و نیز صف فرآیندهای در حال اجرا خالی باشد، سیستم متوقف می‌شود؛ در زمانی که سیستم در حالت متوقف شده است ممکن است که یک یا چند فرآیند در وضعیت بن‌بست قرار داشته باشند؛ سیستم در قبال این فرآیندها هیچ مسئولیتی ندارد.
--	---

شرح مورد استفاده ۱- شرح مورد استفاده Shedule.

Use-Case	Create	
Primary Actor	Process	
Description	این مورد استفاده یک فرآیند جدید ایجاد می‌کند؛ فرآیندی که از این مورد استفاده برای ایجاد فرآیند جدید استفاده کرده است، به عنوان فرآیند والد فرآیند جدید تنظیم می‌شود. همچنین فرآیند والد می‌بایست فرآیندهای Pager و مدیر استثنائات فرآیند جدید را مشخص کند.	
Pre-condition	شرط اولیه برای این مورد استفاده این است که با ایجاد فرآیند جدید تعداد فرآیندهایی که توسط سیستم مدیریت می‌شوند، از تعداد در نظر گرفته شده بیشتر نشود (این مقدار به کوانتم زمانی و کاربرد بستگی دارد).	
Trigger	این مورد استفاده با فراخوان سیستمی createProcess فعال می‌شود.	
Typical Course of Events	Actor Action	System Response
	<p>گام (۱): فرآیند والد شماره شناسایی فرآیندهای Pager و مدیر استثنائات فرآیند جدید را تعیین می‌کند؛</p> <p>گام (۲): فرآیند والد فراخوان سیستمی createProcoss را فرا می‌خواند؛</p>	<p>گام (۳): روال مربوط به فراخوان سیستم فعال می‌شود؛</p> <p>گام (۴): فراخوان سیستم یک فضای آدرس خالی برای فرآیند جدید می‌سازد؛</p> <p>گام (۵): یک شماره شناسایی جدید و منحصر بفرد برای فرآیند جدید انتخاب و شماره شناسایی فرآیند والد، فرآیند Pager، و نیز فرآیند مدیر استثنائات به فرآیند جدید منتسب می‌شود (تنظیم PCB)؛</p> <p>گام (۶): فرآیند جدید در لیست</p>

		فرآیندهای آماده اجرا قرار می‌گیرد؛
Alternative Course:		
Post-condition	بعد از خروج از روال مربوط به فراخوان سیستمی، در صورتی که عملیات موفقیت آمیز باشد، صف برنامه‌های آماده اجرا یک واحد افزایش می‌یابد.	
Business Rules	<ul style="list-style-type: none"> • هر فرآیند برای ایجاد شدن نیازمند یک فضای آدرس خالی می‌باشد؛ • فرآیندهای بعد از ایجاد می‌توانند در صف فرآیندهای آماده اجرا قرار بگیرند؛ • فرآیند Pager مسئول پر کردن فضای آدرس فرآیند جدید می‌باشد؛ • فرآیند والد حق کنترل فرآیندهایی که ایجاد کرده است را دارد، و می‌تواند هر یک از فرزندان خود را خاتمه دهد کند؛ • سیستم عامل هیچ مسئولیتی در قبال معتبر بودن مقادیر تنظیم شده توسط فرآیند والد ندارد. 	

شرح مورد استفاده ۲- شرح مورد استفاده Create.

Use-Case	Abort	
Primary Actor	Process	
Description	فرآیندها می‌توانند با استفاده از این مورد استفاده خود، و یا فرآیندهایی که کنترل آن‌ها را بر عهده دارند را خاتمه دهند. به محض خاتمه یک فرآیند تمامی منابع آن به فرآیندهایی که آن منابع را برایش تامین کرده بودند باز گردانده می‌شود.	
Pre-condition	فرآیند مشخص شده موجود باشد.	
Trigger	این مورد استفاده با فراخوان سیستمی abortProcess فعال می‌شود.	
Typical Course of Events	Actor Action	System Response
	گام (۱): فرآیند درخواست کننده فرآیند مورد نظر را مشخص می‌کند؛ گام (۲): فرآیند درخواست کننده فراخوان سیستمی abortProcش را فرا می‌خواند؛	گام (۳): روال مربوط به فراخوان سیستم فعال می‌شود؛ گام (۴): روال مورد نظر از لیست فرآیندهای سیستم حذف می‌شود؛ گام (۵): تمامی فرآیندهایی که توسط این فرآیند ایجاد شده‌اند، این فرآیند Pager، و یا مدیر استثنائات آن‌ها باشد، خاتمه می‌یابد؛ گام (۶): فضای آدرس و تمامی منابع

		اختصاص داده شده به فرآیند، به فرآیند تأمین کننده بازگردانده می شود؛
Alternative Course:		
Post-condition	در صورت موفقیت آمیز بودن عملیات، فرآیند مشخص شده و تمامی فرآیندهای فرزند این فرآیند خاتمه می یابند.	
Business Rules	<ul style="list-style-type: none"> هر فرآیند تنها زمانی می تواند فرآیندی را خاتمه دهد که: ۱- فرآیند مورد نظر خود فرآیند باشد، ۲- فرآیند استفاده کننده از این فرخوان جزو فرآیندهای Pager، والد، و یا مدیر استثنائات فرآیند مورد نظر باشد؛ فرآیند تأمین کننده منابع برای یک فرآیند، فرآیند Pager آن فرآیند می - باشد. 	

شرح مورد استفاده ۳- شرح مورد استفاده Abort.

Use-Case	Get PID	
Primary Actor	Process	
Description	یک فرآیند می تواند با استفاده از این مورد استفاده شماره شناسائی خود، فرآیند والد، مدیر استثنائات، و نیز Pager را بدست آورد.	
Pre-condition		
Trigger	این مورد استفاده با فراخوان سیستمی getPID فعال می شود.	
Typical Course of Events	Actor Action	System Response
	گام (۱): فرآیند درخواست کننده فراخوان سیستمی getPID را فرا می خواند؛	گام (۳): روال مربوط به فراخوان سیستم فعال می شود؛ گام (۴): سیستم عامل شماره شناسائی خود، فرآیند والد، مدیر استثنائات، و نیز Pager فرآیند جاری را در اختیار فرآیند جاری قرار می دهد؛
Alternative Course:		
Post-condition	بعد از خروج از روال مربوط به فراخوان سیستمی، حالت سیستم بدون تغییر باقی می ماند.	
Business Rules		

شرح مورد استفاده ۴- شرح مورد استفاده Get PID.

Use-Case	Force Schedule	
Primary Actor	Process	
Description	فرایندها می‌توانند با استفاده از این مورد استفاده هر یک از فرایندهایی را که تحت کنترل دارند را از حالت مسدود به حالت آماده اجرا منتقل کنند.	
Pre-condition	شماره شناسایی فرآیند موجود باشد، و فرآیند درخواست کننده کنترل آن فرآیند را در اختیار داشته باشد، و همچنین فرآیند مورد نظر در صف فرایندهای مسدود باشد.	
Trigger	این مورد استفاده در هر بار فراخوانی فراخوان سیستم forceSchedule فعال می‌شود.	
Typical Course of Events	Actor Action	System Response
	گام (۱): فرآیند درخواست کننده، شماره شناسایی فرآیند مورد نظر را مشخص می‌کند؛ گام (۲): فرآیند در خواست کننده فراخوان سیستمی forceSchedule را فراخوانی می‌کند؛	گام (۳): روال مربوط به فراخوان سیستمی فعال می‌شود؛ گام (۴): فرآیند مورد نظر از لیست فرایندهای مسدود حذف و به صف فرایندهای آماده اجرا می‌پیوندد؛
Alternative Course:		
Post-condition	بعد از خروج از روال مربوط به فراخوان سیستم، در صورتی که عملیات موفقیت آمیز باشد یک فرآیند به صف فرایندهای آماده اجرا افزوده می‌شود.	
Business Rules	<ul style="list-style-type: none"> یک فرآیند زمانی می‌تواند فرآیندی را از لیست فرایندهای مسدود شده خارج کند که فرآیند درخواست کننده، فرآیند والد، فرآیند Pager، و یا مدیر استثنائات فرآیند مورد نظر باشد؛ سیستم عامل هیچ تضمینی در مورد رفتار فرآیند خارج شده از لیست فرایندهای مسدود نمی‌کند. 	

شرح مورد استفاده ۵ - شرح مورد استفاده Force Schedule

Use-Case	Reclaim Page
Primary Actor	Process
Description	فرایندها با استفاده از این مورد استفاده می‌توانند بصورت صریح صفحه‌ایی را که قبلاً به فضای آدرسی (هر فضای آدرس متناظر با یک فرآیند می‌باشد) Map کرده بودند، باز پس بگیرند. فرآیند باز پس گیری می‌تواند بصورت زنجیره‌ایی باشد.

Pre-condition	شماره شناسایی فرآیند، موجود باشد.	
Trigger	این مورد استفاده در هر بار فراخوانی فراخوان سیستم reclaimPage فعال می-شود.	
Typical Course of Events	Actor Action	System Response
	گام (۱): صفحه مورد نظر را مشخص می‌کند؛ گام (۲): فرآیند در خواست کننده فراخوان سیستمی reclaimPage را فراخوانی می‌کند؛	گام (۳): روال مربوط به فراخوان سیستمی فعال می‌شود؛ گام (۴): به ازای تمامی فضاهای آدرس موجود در سیستم، صفحه مشخص شده از آنها حذف می‌شود؛
Alternative Course:		
Post-condition	بعد از خروج از روال مربوط به فراخوان سیستم، در صورتی که عملیات موفقیت آمیز باشد، یک صفحه از فضای آدرس یک یا چند جدول فرآیند کم می‌شود.	
Business Rules		

شرح مورد استفاده ۶- شرح مورد استفاده Reclaim Page

Use-Case	Map Page	
Primary Actor	Process	
Description	فرآیندها با استفاده از این مورد استفاده می‌توانند بصورت صریح صفحه‌ایی از صفحات فضای آدرس خود را به فضای آدرس فرآیند مشخص شده دیگری اضافه کنند بطوری که صفحه مشخص شده در فضای آدرس فرآیند درخواست کننده باقی بماند (بدین ترتیب فرآیندها می‌توانند بخش‌هایی از فضای آدرس خود را به اشتراک بگذارند).	
Pre-condition	شماره شناسایی فرآیند موجود باشد، و فرآیند مشخص شده به خاطر درخواست صفحه بلاک شده باشد.	
Trigger	این مورد استفاده در هر بار فراخوانی فراخوان سیستم mapPage فعال می‌شود.	
Typical Course of Events	Actor Action	System Response
	گام (۱): فرآیند درخواست کننده، شماره شناسایی فرآیند مورد نظر و شماره صفحه را که می‌خواهد Map کند را مشخص می‌کند؛	گام (۳): روال مربوط به فراخوان سیستمی فعال می‌شود؛ گام (۴): صفحه مورد نظر را به فضای آدرس فرآیند مشخص شده اضافه می-

	کنیم؛	گام (۲): فرآیند در خواست کننده فراخوان سیستمی mapPage را فراخوانی می‌کند؛
Alternative Course:		
Post-condition	بعد از خروج از روال مربوط به فراخوان سیستم، در صورتی که عملیات موفقیت آمیز باشد یک صفحه به فضای آدرس فرآیند مشخص شده افزوده می‌شود.	
Business Rules	<ul style="list-style-type: none"> یک فرآیند زمانی می‌تواند صفحه‌ایی را به فضای آدرس فرآیندی دیگر نگاشت کند که آن فرآیند به دلیل Page Fault مسدود شده باشد؛ فرآیند نگاشت کننده در هر زمان می‌تواند صفحه نگاشت شده را بازپس بگیرد. 	

شرح مورد استفاده ۷- شرح مورد استفاده Map Page.

Use-Case	Grant Page	
Primary Actor	Process	
Description	فرآیندها با استفاده از این مورد استفاده می‌توانند بصورت صریح صفحه‌ایی از فضای آدرس خود را به فضای آدرس فرآیند مشخص شده دیگری اضافه کنند؛ این اضافه کردن به گونه‌ایی است که صفحه مشخص شده از فضای آدرس فرآیند اعطا کننده حذف می‌شود.	
Pre-condition	شماره شناسایی فرآیند موجود باشد، و به خاطر Page Fault مسدود شده باشد.	
Trigger	این مورد استفاده در هر بار فراخوانی فراخوان سیستم grantPage فعال می‌شود.	
Typical Course of Events	Actor Action	System Response
	گام (۱): فرآیند درخواست کننده، شماره شناسایی فرآیند مورد نظر و شماره صفحه را که می‌خواهد Grant کند را مشخص می‌کند؛	گام (۳): روال مربوط به فراخوان سیستمی فعال می‌شود؛
	گام (۲): فرآیند در خواست کننده فراخوان سیستمی grantPage را فراخوانی می‌کند؛	گام (۴): صفحه مورد نظر را به فضای آدرس فرآیند مشخص شده اضافه می‌کنیم؛
		گام (۵): صفحه مورد نظر را از فضای آدرس فرآیند اعطا کننده حذف می‌کنیم؛
Alternative Course:		
Post-condition	بعد از خروج از روال مربوط به فراخوان سیستم، در صورتی که عملیات	

	موفقیت آمیز باشد یک صفحه به فضای آدرس فرآیند مشخص شده اضافه، و نیز یک صفحه از فضای آدرس فرآیند درخواست کننده حذف می شود.
Business Rules	<ul style="list-style-type: none"> • یک فرآیند زمانی می تواند صفحه ای را به فضای آدرس فرآیند دیگر اعطا کند که آن فرآیند به دلیل Page Fault مسدود شده باشد؛ • فرآیند اعطا کننده دیگر نمی تواند صفحه اعطا شده را بازپس بگیرد.

شرح مورد استفاده ۸- شرح مورد استفاده Grant Page.

Use-Case	Send Message	
Primary Actor	Process	
Description	فرایندها می توانند به وسیله این مورد استفاده پیام هایی را به فرایندهای دیگر ارسال کنند. در صورتی که فرآیند مشخص شده منتظر رسیدن پیامی باشد، فرآیند از حالت مسدود به حالت آماده اجرا تغییر وضعیت می دهد.	
Pre-condition	شماره شناسایی فرآیند موجود باشد، و نیز منتظر رسیدن پیام از سوی یک فرآیند باشد.	
Trigger	این مورد استفاده در هر بار فراخوانی فراخوان سیستم sendMessage فعال می شود.	
Typical Course of Events	Actor Action	System Response
	گام (۱): فرآیند درخواست کننده، شماره شناسایی فرآیند مورد نظر و پیام مورد نظر را مشخص می کند؛ گام (۲): فرآیند در خواست کننده فراخوان سیستمی sendMessage را فراخوانی می کند؛	گام (۳): روال مربوط به فراخوان سیستمی فعال می شود؛ گام (۳): پیام مشخص شده به دست فرآیند مورد نظر می رسد و فرآیند از حالت مسدود به آماده اجرا تغییر وضعیت می دهد؛
Alternative Course:		
Post-condition		
Business Rules	سیستم عامل هیچ تضمینی در مورد به نتیجه رسیدن پیام ارسالی نمی کند.	

شرح مورد استفاده ۹- شرح مورد استفاده Send Message.

Use-Case	Receive Message
Primary Actor	Process
Description	فرایندها می توانند به وسیله این مورد استفاده پیام هایی را از فرایندهایی دریافت کنند. در صورتی که فرآیندی از این مورد استفاده، استفاده کند بدون توجه به هیچ چیز، تا زمان رسیدن پیام از منبع مشخص شده به حالت مسدود وارد می -

	شود.	
Pre-condition		
Trigger	این مورد استفاده در هر بار فراخوانی فراخوان سیستم receiveMessage فعال می شود.	
Typical Course of Events	Actor Action	System Response
	گام (۱): فرآیند درخواست کننده، شماره شناسائی فرآیند مورد نظر را مشخص می کند؛ گام (۲): فرآیند در خواست کننده فراخوان سیستمی receiveMessage را فراخوانی می کند؛	گام (۳): روال مربوط به فراخوان سیستمی فعال می شود؛ گام (۴): فرآیند در خواست کننده به خاطر انتظار برای پیام از سوی فرآیندی دیگر مسدود می شود؛
Alternative Course:		
Post-condition		
Business Rules	سیستم عامل هیچ تضمینی در مورد خروج فرآیند درخواست کننده از وضعیت مسدود نمی کند.	

شرح مورد استفاده ۱۰- شرح مورد استفاده Receive Message

Use-Case	Dispatch	
Primary Actors	Exception, Interrupt	
Description	در این مورد استفاده وقفه و یا استثناء ایجاد شده به برنامه مورد نظر ارسال می شود. برای هر وقفه و یا به ازای هر برنامه یک مدیر وقفه و مدیر استثنائات تعریف شده است.	
Pre-condition	برنامه مدیر وقفه و یا مدیر استثنائات باید منتظر دریافت پیام باشد، در غیر این صورت وقفه و یا استثناء بدون پاسخ خواهد ماند.	
Trigger	وقفه سخت افزاری، و یا استثناء.	
Typical Course of Events	Actor Action	System Response
	گام (۱): وقفه سخت افزاری ایجاد می شود؛	گام (۲): بخش مربوط توزیع وقفه مورد نظر فعال می شود؛ گام (۳): در صورتی که فرآیندی برای مدیریت این وقفه منتسب شده باشد پیامی مبنی بر وقوع وقفه به برنامه مورد نظر

		ارسال می‌شود؛ گام (۴): با خروج از بخش توزیع وقفه‌ها، سیستم به کار خود ادامه می‌دهد.
Alternative Course:	گام آلترناتیو (۱): یک استثناء توسط برنامه جاری ایجاد می‌شود.	گام آلترناتیو (۲): بخش مربوط به توزیع استثنائات فعال می‌شود؛ گام آلترناتیو (۳): برنامه جاری بلوک می- شود؛ گام آلترناتیو (۳): پیامی مبنی بر بروز یک استثناء به برنامه مدیر استثنائات منتسب به برنامه مورد نظر ارسال می‌شود.
Post-condition	بعد از خروج از زمانبند، بخش توزیع وقفه‌ها یا استثنائات یک پیام به برنامه‌ای منتسب به یک استثناء با وقفه ارسال می‌شود؛ در صورتی که از بخش مدیریت استثنائات خارج شویم، فرآیند جاری به حالت بلوک شده در می‌آید.	
Business Rules	<ul style="list-style-type: none"> • در صورتی که برنامه‌ای مسئول استثنائات برنامه‌ای دیگر نباشد و یا اینکه در حین بروز استثناء منتظر دریافت پیام نباشد، سیستم عامل هیچ مسئولیتی در قبال فرآیند بلاک شده ندارد؛ • در صورتی که هیچ فرآیندی مسئول مدیریت وقفه‌ها نباشد و یا اینکه در هنگام بروز وقفه هیچ فرآیندی منتظر دریافت پیام نباشد، سیستم عامل هیچ مسئولیتی در قبال وقفه بدون پاسخ ندارد. 	

شرح مورد استفاده ۱۱- شرح مورد استفاده Dispatch

۴-۲- قواعد حاکم بر سیستم

در این بخش قصد داریم برای اینکه بتوانیم با دقت بیشتر سیستم خود را صحت سنجی کنیم، برخی از قواعد حاکم بر سیستم (یا Business Rule ها) را بصورت لیست وار ذکر کنیم. البته لازم به ذکر است که این قواعد را در ادامه روند طراحی می‌توانیم بهتر و با دقت و جزئیات بیشتری بیان کنیم.

در حالت کلی و در یک بررسی اولیه می‌توان گفت که باید این قواعد بر سیستم ما حاکم باشد (لازم است بدانیم این قواعد شرایط ثابت سیستم را بیان می‌کنند به عبارتی دیگر در کل زمان فعالیت سیستم، این شرایط برقرار خواهند بود):

- هر فرآیند دارای یک و تنها یک فضای آدرس است.
- در سیستم تعداد محدودی (ثابت و غیر صفر) فرآیند وجود دارد.
- هر فرآیند باید دارای یک شناسه منحصر بفرد باشد.
- در هر فضای آدرس تعداد نگاشت‌ها حداکثر برابر یک مقدار ثابت و غیر صفر است.
- در سیستم تنها سه صف برای مدیریت فرآیندها وجود دارد (صف‌هایی با نام های ready, running, و blocked که به ترتیب فرآیندهای آماده اجرا، در حال اجرا، و بلاک شده را در خود نگه می‌دارند).
- در هیچ کدام از سه صف سیستم تکرار وجود ندارد.
- طول صف فرآیندهای در حال اجرا همواره کمتر یا برابر با یک می‌باشد، چرا که این سیستم عامل برای سیستم‌های تک پردازنده‌ای طراحی می‌شود.
- یک فرآیند در تمام مدت حیات خود در هر لحظه فقط در یکی از سه صف سیستم قرار خواهد داشت.
- در هر لحظه در این سه صف سیستم چیزی جز تمامی فرآیندهای ایجاد شده تا آن لحظه وجود ندارد.
- در هر لحظه حداقل یک فرآیند برای اجرا شدن وجود دارد (صف‌های ready و running با هم و بطور همزمان صفر نمی‌شوند).

- در ابتدا فرآیندی با نام `root_server` در سیستم در حال اجرا می‌باشد (فرض بر آن است که این فرآیند تمامی فضای آدرس فیزیکی سیستم را در اختیار دارد و تمامی فرآیندها را این فرآیند ایجاد می‌کند).
- همواره فرآیند `root_server` در سیستم حضور دارد.

همانطور که ملاحظه می‌شود این قواعد در این مرحله خیلی غیر دقیق و زبانی بیان شده‌اند، و حتی برخی از آن‌ها نیز بیان نشده‌اند؛ در ادامه با روشن‌تر شدن نیازمندی‌ها و قدری پالایش (Refinement) آن‌ها، قادر خواهیم بود این قواعد را که همواره در سیستم ما برقرار خواهند بود بطور خیلی دقیق‌تر بیان کنیم. بعد از آن می‌توانیم با استفاده از عبارت‌های OCL و روش‌هایی که در فصل قبل ارائه شد، این قواعد را بازنویسی کرد؛ بدین ترتیب قادر خواهیم بود مدل UML خود را بدون ابهام به زبان فرمال AMN ترجمه کنیم.

۴-۳- نتیجه‌گیری

در این فصل نیازمندی‌های سیستم را بیان کردیم، این نیازمندی‌ها را با استفاده از دیاگرام موارد استفاده به تصویر کشیدیم و در بخش شرح موارد استفاده توضیح کامل هر یک از موارد استفاده را آوردیم. در انتها نیز قواعد حاکم بر سیستم را داشتیم که می‌تواند بصورت گزاره‌های همواره ثابت در سیستم اعمال شوند.